# Executive Summary

## Who should read this report?

# 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.

## Managing projects

## 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 efficiency.

### UCM, base ClearCase – automating and integrating change management process

Change management is an essential function of software project management. The ability to modify, store and deliver software according to the changing needs of the project requirement is critical for project success.

ClearQuest is the de facto tool within Rational™ suite to provide change management. However, ClearCase uses change management models to facilitate integration with ClearQuest.

UCM (Unified Change Management) is the out-of-box change management process model. UCM is activity based; project assets that are changed during a change activity are mapped and audited accordingly.

Base ClearCase is a term that refers to a set of tools within ClearCase to configure a tailor-made change management solution that suits the need of the project.

Both UCM and base ClearCase functions as an effective change management model that can be further used and integrated with other tools.

### VOB, Snapshot and Dynamic Views – automating workspace management

Workspace is an abstract collection of project assets and resources that facilitates individuals to work as a cohesive unit.

Workspace enables an individual developer with the ability to work and later on integration his work with others.

ClearCase takes a “Sandbox” approach when dealing with workspace management.

VOB (Versioned Object Base) is ClearCase’s terminology for a software repository. Developers use the check-in and check-out functions to interact with VOBs. VOB keep track of all versioned assets and auditing information.

To put it simply, the major advantage of using ClearCase *View* feature is that it guarantees project consistency and stability.

Each *View* maintains and keeps its own changes; developers working with other *Views* are not affected by changes that occurred within a particular *View*. However, they may access other views provided they have the appropriate access permissions.

ClearCase View is available as Snapshot and Dynamic modes. In Snapshot model, ClearCase *View* retrieves and copies all items that are listed in the configuration specification of the project.

Dynamic mode uses ClearCase’s MVFS (multi-version file system) and facilitates immediate and transparent access to data in the repository.

ClearCase *View* automates workspace management by providing version selection, source retrieval and knowledge sharing.

### Baseline - automating the creation process

In ClearCase, a baseline is a set of versioned project assets that is consistent with a particular project mile stone.

An example of a project baseline would consist of all project source code files of beta 0.1 and possibly also including alpha 0.2’s forked changes.

ClearCase provides baseline function for both the UCM and base ClearCase interfaces.

The important distinction between the two baseline functions is that base ClearCase requires the developer to configure *config spec* of his workspace (view); *config spec* is a set of rules that describes and controls the versioned project assets. An example of a *config spec* rule would be the requirement that all source code of module 101 should only include code version 0.11.

UCM do not require individual developer to do this, versioned specification and related requirements of the project baseline are configured by the maintainer or manager of the project prior to usage.

Using the baseline automates the development creation process. By providing all team members the right set of versioned assets; the project management ensures the consistency and stability of the project code base.

### Automating software build process

Space and time restrictions are common to all software engineering projects. Traditional version control systems such as CVS implements *source* control but do not automate the software building process.

A scalable and reasonably complex project would undergo many changes and releases during of the duration. Different branching within the project with different configuration of features would require different builds for release.

We needed a tool to automate this process; the alternative is using a manual and ad hoc process where we would manually link and combine source files for compilation. The main functional requirement of this automation is build avoidance; or in laymen terms: avoid building/compiling unchanged part of the project assets.

In ClearCase, the function of automating software building is accomplished by using the tool clearmake and omake.

clearmake is ClearCase’s implementation of the popular UNIX utility *make* with improved features.

omake is another tool which is unique to ClearCase. It is best used in the dynamic view interface.

## Audit Trails

## Integration with Rational™ Suite and popular IDEs

## Collaboration

Support for parallel development

Multi-platform support

Implementing and enforcing project policies

# References

ClearCase Concepts Guide

http://techpubs.sgi.com/library/dynaweb\_docs/0620/SGI\_Developer/books/ClrC\_CG/sgi\_html/ch01.html

ClearCase Usage Tutorial

https://publib.boulder.ibm.com/infocenter/cchelp/v7r1m0/index.jsp?topic=/com.ibm.rational.clearcase.tutorial.doc/topics/default\_topic.htm