



# **COLLABORATIVE LIFECYCLE MANAGEMENT (CLM)**

**Vidhya V Kumar**



# THE BASIC QUESTIONS

Collaboration between ...?

Lifecycle of/ for ...?

Management of ...?



# SOFTWARE DELIVERY LIFECYCLE

All steps from concept to creation

- Analyze requirements
- Develop code
- Test code
- Release

Traditionally, these have been called stages because each requires specific non-fungible skills

Modern agile models compress these stages into ‘runs’ or ‘sprints’ with each sprint comprising of all stages.

‘Deployment’ and ‘documentation’ are two activities that cut across most of the lifecycle. There are others.



# MANAGEMENT TASKS

Software delivery requires several management tasks

## Example

- Work assignments
- Work scope
- Schedules
- Dependencies between various activities
- Workflows

These activities are highly interdependent.

And therefore, we need a tool to manage them.



# COLLABORATION BETWEEN ...

Not all the activities in a delivery lifecycle are managed by one team.

The number of teams and their tasks depend on the delivery model.

A waterfall model for instance will have teams based on lifecycle stages.

A SAFe model could have small sprint teams that contain developers, testers, operations personnel, and even documentation personnel.

Interdependent activities lead to interdependent teams (though the nature of dependence depends on the delivery model).

Whatever be the model, collaboration is a must.

**That is the management of lifecycle activities has to be collaborative.**

Hence the name **Collaborative Lifecycle Management or CLM.**



# SO WE HAVE CLM

CLM is a tool that helps software delivery teams performing various lifecycle activities to manage their activities collaboratively



# THE LIFECYCLE DOMAINS AND OTHER CATEGORIES

In CLM, we term the various categories of lifecycle activities, domains

The following domains are referenced by CLM:

- Requirements analysis
- Quality (testing)
- Design

These ‘lifecycle delivery activities’ need to be managed



# CLM ACTIVITY CATEGORIES

CLM helps manage these lifecycle delivery activities

CLM classifies management functions as follows:

- Requirements management
- Design management
- Quality management

There are also cross-domain management functions, viz:

- Project management
- Configuration management
- Reporting



# THE CLM SUITE

Each of these management functions is mapped to an 'application' within the CLM suite

Requirements management	<ul style="list-style-type: none"><li>• Rational Doors Next Generation</li></ul>
Design management	<ul style="list-style-type: none"><li>• Rational Rhapsody</li><li>• Design Management for Rational Software Architect</li></ul>
Quality management	<ul style="list-style-type: none"><li>• Rational Quality Manager</li></ul>
Project management	<ul style="list-style-type: none"><li>• Rational Team Concert</li></ul>
Configuration Management	<ul style="list-style-type: none"><li>• Rational Team Concert</li><li>• Configuration Manager: Local and Global</li></ul>
Reporting	<ul style="list-style-type: none"><li>• Jazz Reporting Service</li></ul>



# CLM SUITE USAGE

CLM is a suite of applications.

Not all the applications are installed at every customer site.

Also the suite has evolved over the years.

Therefore there is some amount of overlap in capability (functionality) between these applications.

This is evident in the documentation as well.

More on that later ...