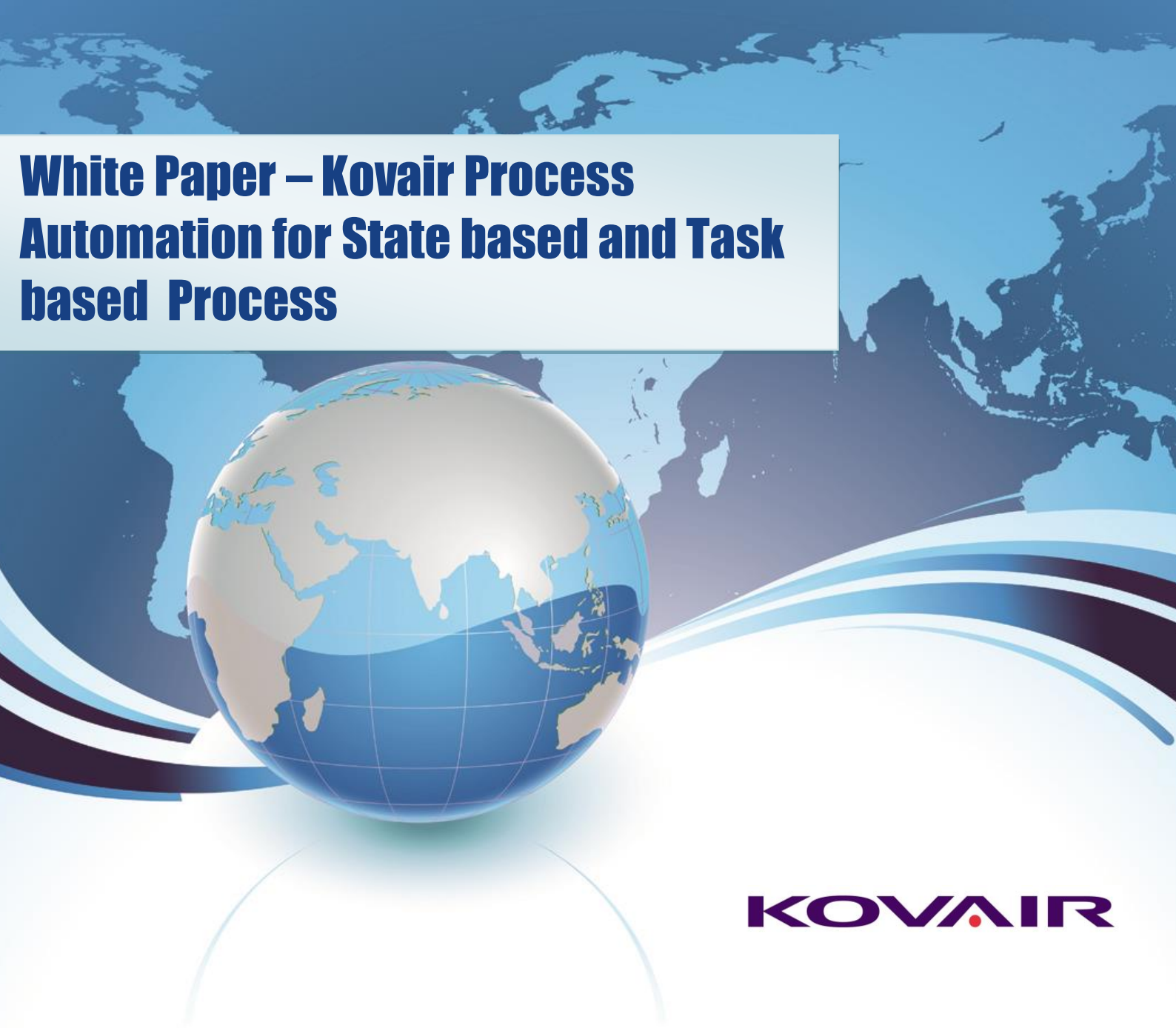
White Paper – Kovair Process Automation for State based and Task based Process



KOVAIR

By Kovair Marketing

US Corporate Office

Kovair Software, Inc.
2603 Camino Ramon,
STE 200, San Ramon,
CA 94583, United States
Tel: 1.408.262.0200 Extn.1
Email: sales@kovair.com

India Registered Office

Kovair Software Pvt. Ltd.
PTI Building, 6th Floor, DP-9,
Sector – V, Salt Lake City,
Kolkata - 700091, India
Tel: 91-33-4065 7016/17/18/19
Email: sales@kovair.com

Bangalore Office

Kovair Software Pvt. Ltd.
Samad House, #402, 4th B cross,
7th A Main, HRBR Layout, Kalyan
Nagar, Bangalore- 560043, India
Tel: +91-95350 92589
Email: sales@kovair.com

Table of Contents

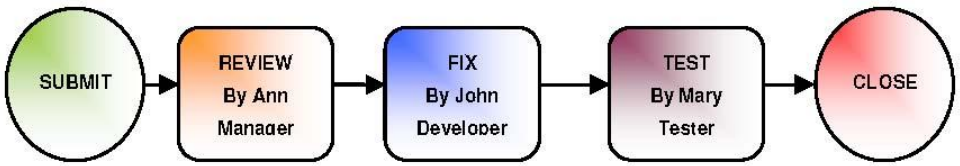
Introduction

Kovair ALM Studio is a Software Development Lifecycle Process Management system. Every organization follows processes which are appropriate for a project, and incorporate the culture of the group and other variables. Kovair ALM Studio helps organizations implement processes whether it is a simple State-based one or something that is much more advanced (viz. RUP, Extreme Programming etc). Kovair's visual Process Designer makes it easy to visualize, design and implement development processes of any complexity; be it for a Project, Release, Module, Requirement or Issue.

In this whitepaper, we will briefly describe how Kovair supports and automates State-based processes. We will also describe in detail Kovair's powerful Process Automation system based on Tasks. This Task-based process can be used to automate any development process, particularly those where State-based processes fall short.

State-based Process

A State-based Process depends on two variables for each artifact (a Requirement or an Issue) – a 'State' and an 'Owner'. By changing the State from one value to another the artifact moves along in the Process. As it changes the State often it is assigned to a different person, the "Owner", who will be responsible for that Item at that State. A simple example of an Issue Process is shown below.



Here, the different Status values are **SUBMIT**, **REVIEW**, **FIX**, **TEST** and **CLOSE**. The users are *Ann Manager*, *John Developer* and *Mary Tester*. This is the way it works:

When an Issue is created the default State is 'Review' and it is assigned to *Ann Manager* as owner. Once *Ann Manager* has reviewed the Issue she can decide whether it is really an Issue, in that case she changes the Status to 'Fix' and Owner to '*John Developer*'. *John Developer* may get an email notification that an Issue has been assigned to him. After he fixes the Issue he changes the Status to 'Test' and The Owner to *Mary Tester*. After testing, if *Mary Tester* finds it is fixed, she can change the Status to 'Closed', and the Issue is resolved. Otherwise, she can change the Status to 'Fix' again and assign IT back to '*John Developer*'.

From this simple Process the following characteristics of a State-based Process can be seen:

- For any State there can be exactly one Owner
- At any State, the Owner needs to know what are the next possible States and who their respective Owners are
- It is a manual process whose success depends on the discipline and memory of each Owner at each State. If Ann changes the Status to 'Fix' and mistakenly sends it to Mary Tester, the Process breaks down or at best needs someone to reroute it to the right person

For the above reason, a State-based manual Process can be used ONLY for small teams that are located in one place and where everyone knows exactly what the others are doing.

Kovair not only supports State-based Processes, but automates them too. By automating a State-based Process, Kovair can make it error-proof and applicable to bigger teams and more complex processes. Users can implement State-based Processes for Projects, Releases, Modules, Requirements, and Issues.

Kovair Policy Engine

In the Policy Engine the Administrator can define different Rules for automating a process. Each Rule has a Condition and an Action. A Condition can have multiple clauses as well as Actions.

EXAMPLE:

Condition: 'If an Issue is submitted' and 'Issue has a high priority'.

Action: 'Set Status to Review' and 'Set Owner to Ann manager' and 'Send Notification to Ann Manager'.

Similarly there can be another Policy with

Condition: 'If Status is changed to Fix'

Action: 'Set Owner to John Developer' and 'Send Notification to John Developer'.

In this case, Ann just needs to change the Status to Fix and need not know who the person responsible for fixing this Issue is. In fact, the last Policy can be made even more flexible. Say, there are two developers John and Bob. All Issues which belong to the 'Database' module are fixed by Bob and all other Issues are fixed by John. To automate such a rule, a Kovair Policy can be defined as below:

CONDITION DEFINITION SCREEN FOR POLICY

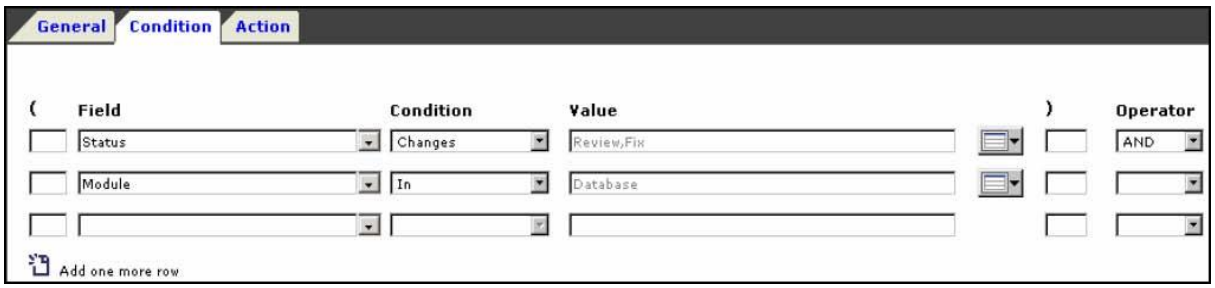


Fig: Condition Definition Screen for Policy

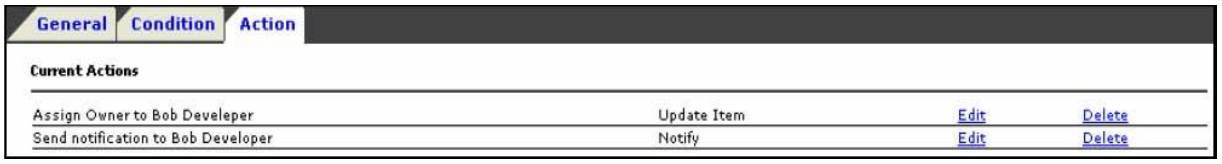


Fig: Action Definition Screen for Policy

Policy #1:

Condition: 'If Issue belongs to Database Module' and 'If Issue Status is changed From Review to Fix'

Action: 'Set Owner to Bob Developer' and 'Send Notification to Bob Developer'

Policy #2:

Condition: 'If Issue does not belong to Database' and 'If Issue Status is changed From Review to Fix'

Action: 'Set Owner to John Developer' and 'Send Notification to John Developer'

Using Kovair Policy Engine one can create a flexible State-based Process and automate it. However, State-based Processes still have their limitations. For this and other reasons, Kovair has developed a powerful Process: 'Task-based Process Engine'.

Limitations of State Based Process

- A State-based Process does not allow multiple owners at a State. In real life, processes often need more than one owner at a single State. A typical example is: 'Review a Requirement' may be assigned to multiple persons, or, in Extreme Programming, each Develop/ Test state is assigned to a pair of developers.

- The current Owner always needs to know what are the different next States available and has to choose them manually based on some policy. For a linear process this may seem simple, but for a more complex process this may become a burden on users as they will need to memorize these policies. For example for a process the policy may be 'If the Issue is a Change

Request and the priority is High and belongs to the Database Module change the Status to Database Review' else 'If the Issue is a Defect and belongs to the UI Module change the Status to Design' etc. Since the Status remains the trigger by which the Process actually moves, users need to understand the complete Process in terms of legal Status changes. This is often the weakest link in a State-based Process and the primary cause of failure.

- However, Kovair Policy Engine ELIMINATES the need to memorize by having the System memorizes, and automates the process. Owners of a State need not know what the next State needs to be.

- There is no inherent way to record the progression of the Process and no way to measure a State within a State. For example, John Developer has no way to communicate that he has started fixing the Issue and he is in the 'Work In Progress' State within the Fix Status.

- The only way to communicate with the Owner is by email notification which is typically outside the control of a Process Management system.

Introduction to Kovair Task-based Process

Kovair has developed a Task-based Process Engine to address the limitations of State-based Processes and especially to make a Process successful in a distributed environment. Unlike a State-based Process, there can be multiple assignees for an Artifact (Requirement or Issue) at any point of time and each assignee gets a Task with an Activity.

EXAMPLE:

In a Requirement Process, a Use Case may be worked on simultaneously by the QA Engineer on a 'Test case design', by the Architect on a 'System design' as well as by the Technical Writer on 'Documentation'.

All these activities will be going on in parallel. Each of these assignees gets a Task in their Home page instructing them to do the particular activity. Once the Task is complete the user just closes the Task. At this juncture, the Kovair Process will automatically generate the next Task(s) for the appropriate persons (based on the defined Process). It is not necessary for any of the participants to know neither what the next steps in the Process are nor whom to assign it to next. This is extremely useful in a larger team especially when they are distributed in multiple locations.

Kovair allows not just one process but five different types of processes running simultaneously at different levels:

- Project level Process – Exactly one instance running for the Project
- Release level Process – Multiple Instances running for each Release

- Module level Process - Multiple Instances running for each Module
- Issue Process - Multiple Instances running for each Issue
- Requirement Process - Multiple Instances running for each Requirement

Following are the key elements in Kovair Task based Process:

1. Roles

The main players in a Kovair ALM process are called **Roles**. Examples of Roles are: 'Project Manager', 'Developer', 'QA Tester', 'Customer', 'Tech Lead', 'Product Manager' 'Sponsor'… In a Process, the same person (Kovair User) may play multiple Roles, e.g. Bob is a Developer for the Database Module but a Tester in UI module.

Similarly, multiple people may play the same role e.g. Bob and Mary both are Developers for the Database and UI modules. In Kovair, the determination as to who is the Developer is automatic and based on whether the Issue belongs to the Database or to the UI module.

Assigning multiple persons to a single Role is another flexible way to model a real-life process. There are a few different scenarios where this feature comes in very handy.

EXAMPLE: XP Pair Programming

In Extreme Programming (XP) one of the more interesting components is pair-programming where two developers play the roles of Developer and Tester interchangeably for a single development task. They both are responsible for completing the task. Kovair's 'One Task for All' multiple-role assignment policy allows multiple users to get individual tasks which all refer to the same task and any one can close this task. Using this feature, the pair of developers (in XP) get their individual Tasks and any of them can close it when completed.

EXAMPLE: Load Balancing by FIFO (First In, First Out) assignment

In many organizations each individual resource from a pool accepts one task at a time as they arrive in a stream. The typical example is a pool of support engineers fielding support-related tasks or a pool of developers fixing bugs as they arrive from the QA cycle. Kovair's 'Queued' Tasks allow the system to generate multiple copies of the same Task for all the pool members and when any one of them accepts the Task, all other copies are removed from the other pool members.

EXAMPLE: Independent Review by multiple persons

This is a typical scenario when a group of persons are asked to do independent tasks of the same nature – say reviewing a document and posting individual comments. In Kovair, a quick way is to create a Role called 'Reviewers' and assign all the reviewers to that role. Based on the 'Individual

Task' policy, Kovair creates independent Tasks for all the assignees and they can close their Tasks independent of each other. Moreover, the process designer can specify a policy of how and when the process moves to the next step, say, when 80% of the individual tasks (quorum) are completed or alternatively, when exactly 5 of them are completed.

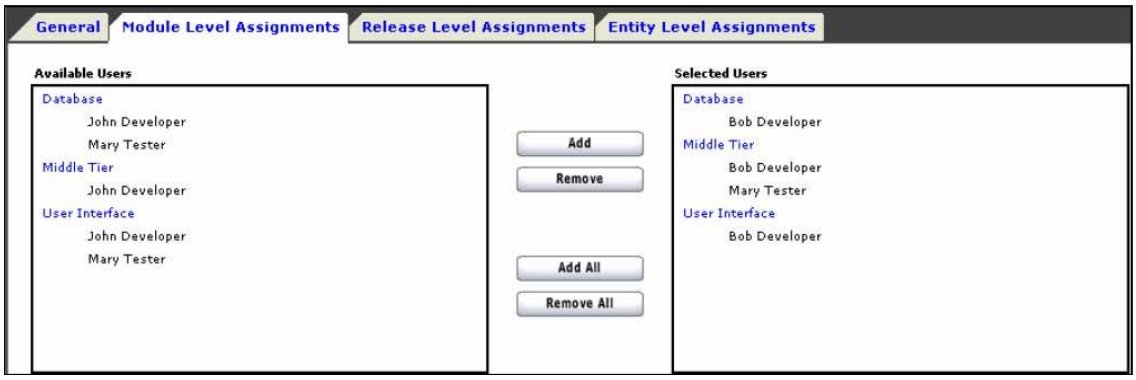


Fig: Multiple Assignments to a Role for Modules

In the Sample Schematic Process the Human Figures designate the Roles. Please note that there are 'Green' Roles and the 'Brown' Roles. The difference between the Green and Brown are: the Brown Role is a Hard Role as defined by YOU. On the other hand, a Green Role is a System-defined Role as anyone can play this Role. For example, in the sample schematic a 'Requirement Proposer' can be anyone and it is automatically assigned to the person whoever has created the new Requirement. If you wish to use the same Proposer in your Process uses the Green Role (and Kovair will know who the person is playing that Role for the particular Requirement). You can have as many Roles as your process demands with whatever names you wish.

2. Steps

Steps help break up a long complex process into modular elements with lesser complexity and duration. It also serves the purpose of high-level description of your process. For example, your Issue Resolution process may have the Steps: 'Review', 'Development' and 'Test'. At any point of time a Requirement or an Issue can belong to exactly one Step. Steps allow you to answer the questions like 'How many Issues are in the Development Step?' or 'What percentage of Requirements are in Review Step?' You can have as many Steps as your process demands with whatever names you wish. In the Sample Schematic Process, the large Gray rectangles designate the Steps.

3. Activities

The atomic units in a Process are called Activities. Activities can be nested within a Step (and can be performed by a Role). Examples of an Activity are: 'Review', 'Implement', 'Fix', 'Test'. In a Step, the Activities can be sequential or parallel and connected to each other by a Connection. As part of the Process execution, when Kovair comes to an Activity, it creates a Task for the user who is assigned to

the particular Role. You can have as many Activities of your choice in each step. In the Sample Schematic Process the small Light Gray rectangles with line attached designate the Activities.

4. Join Nodes

In a Process, often multiple parallel branches merge together in order to go forward. The node at which the connectors merge is called a **Join Node**. It is also referred to as a 'Merge Node', 'Synch Node' or 'Rendezvous Node'. In the Join Node it is possible to define a **Forwarding Policy** which tells the Process when to move forward. Kovair allows two types of Forwarding Policies – Count or Percentage based.

EXAMPLE:

If there are 10 different paths coming to a Join Node from 10 Reviewers of a particular type of Requirement, the Process Designer can say go forward only when Count is 10, which means wait till all the Reviewers complete their review task.

Alternatively he can also specify go forward when Percentage is 60, which means if at least 6 of the reviewers complete their tasks the process can move forward. The last policy is also called quorum-based policy.

5. Wait Nodes

Kovair ALM Studio allows five different types of processes to run simultaneously. They are: Project level Process, Release level Process, Module level Process, Issue Process, and Requirement Process. Kovair ALM Studio also allows synchronization between these processes to model real-life process requirements. An example of this: The interdependencies that exist between a Release (or, for that matter a Project) and all the Requirements & Issues which are planned for that Release. An organization may have a Release process which says, 'unless all the individual Issues are Fixed/ Unit Tested' AND 'all the individual Requirements are developed/ Unit Tested' we CANNOT go into an 'Integration Testing' step of the Release process. Kovair ALM Studio allows the Release Process to have a **Wait Node** which will wait for all the relevant Requirements and Issues to satisfy certain criteria (in this case, pass their own 'Unit Tested' step). It will continuously check for this criterion to be satisfied and ONLY when it does will the Release move to the 'Integration Testing' step by creating a task for the appropriate role, say for the 'Integration Tester'. It is important to note here that it is not necessary to have the individual Requirements and Issue processes to be in place to utilize this handy feature. Even when Requirements & Issues follow a manual, State-based Process (by manually changing a Status field) the Release process can still utilize this synchronization method.

6. Delay Node

A Delay Node allows a Process to pause for a predetermined amount of time (in Hours, Days, Weeks…) or till a date before it proceeds to the next step. This is useful for a Process where certain steps can be traversed only after a particular date or a minimum time is to be given between two steps. It also works very well in escalating issues that are not dealt with on time.

7. Variables in a Process

To control the flow of a Process, any system or custom field of the entity (Requirement or Issue) as well as Project, Release or Module may be used. In addition, a Process may include Process Variables that have the limited scope of the Process only. Any of these variables may be used in the Step, Activities or in any of the other Node types.

8. Conditional Branching

Very few real-life Processes are simple, linear and sequential. Often, Processes follows alternative paths based on different criteria. Kovair ALM Studio allows you to add **Conditional Branching** both between Steps (and between Activities in a Step). The condition can be simple or complex by using AND, OR, NOT and multiple conditional statements. Typically in one or more Activities before the Branching Condition one or more **Variables** are set by the Roles during the execution of the corresponding Tasks. And the Conditions in the Branching are expressed in terms of those variables. In the Sample Schematic Process, the yellow diamond shows the Conditional Branching. The variables used in the Conditional Branching are shown in the Activities (with their possible values in italics). You can have two or more branches out of a Conditional Branch based on different criteria.

9. Rules on Entry and Exit

To make the Processes even more flexible and powerful, Kovair ALM platform has provision to define rules consisting of Condition and Action for entry and exit of every node. The rule definition is similar to the Policy Engine described earlier in this whitepaper. A typical example of this feature is: Notify the Project Manager at Exit of every step in the Release Process (so that he knows exactly what is happening for a Release).

10. Automatic Layout

One of the most useful features of the Process Designer is its ability to automatically do the layout of nodes. This smart tool can be used to focus on selected items as well as on the complete drawing. Rarely does one need to reposition nodes after the automatic layout has been done. This is a life-saver especially when designing large complex processes.

11. Online Documentation

Another useful feature of the Process Designer is the documentation of a Process along with the functional design. It is not necessary to create a separate MS Visio like drawing to document a complex process. To aid in the documentation, Kovair ALM Studio provides tools like **Text Label** and **URL** connected to any node or connector (**Static Text** and **Multi Row/Column Table**), thereby generating documentation on the fly.

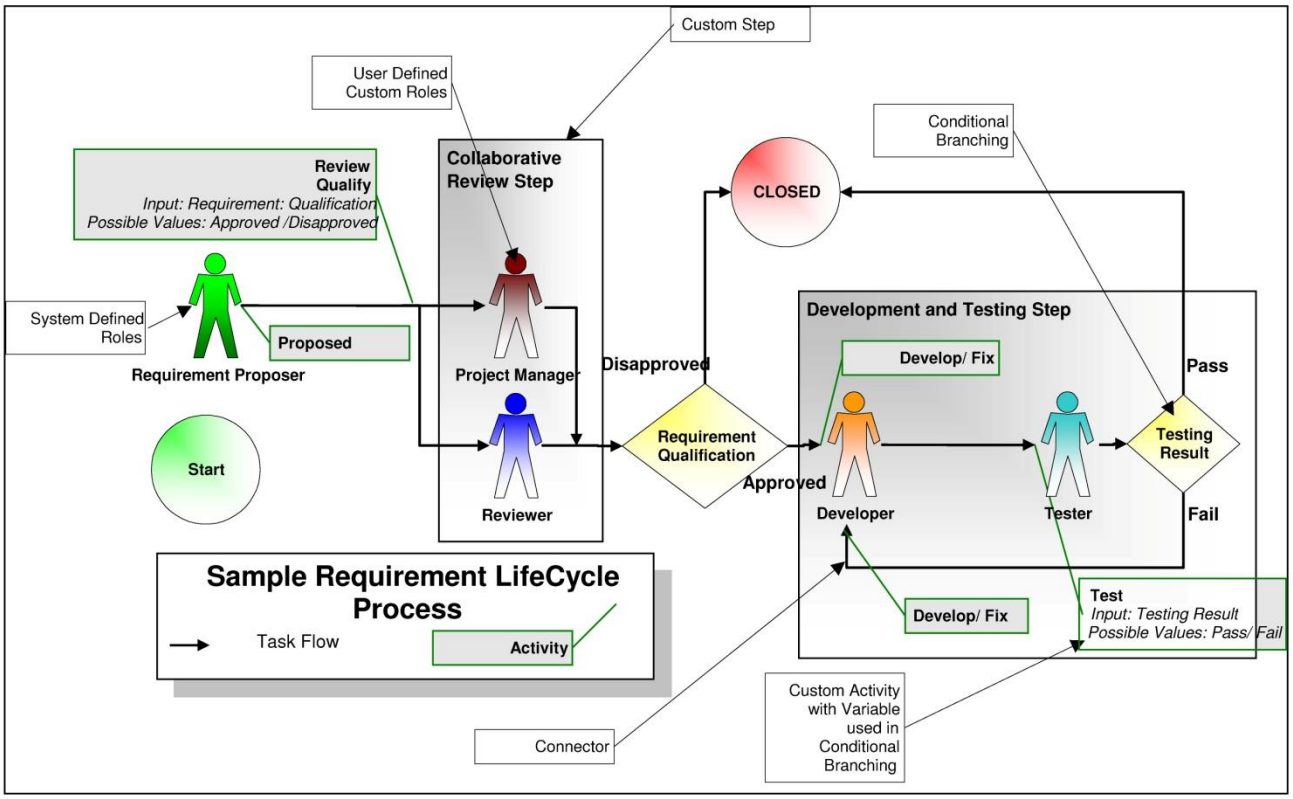


Fig: Schematic of Sample Requirement Lifecycle Process

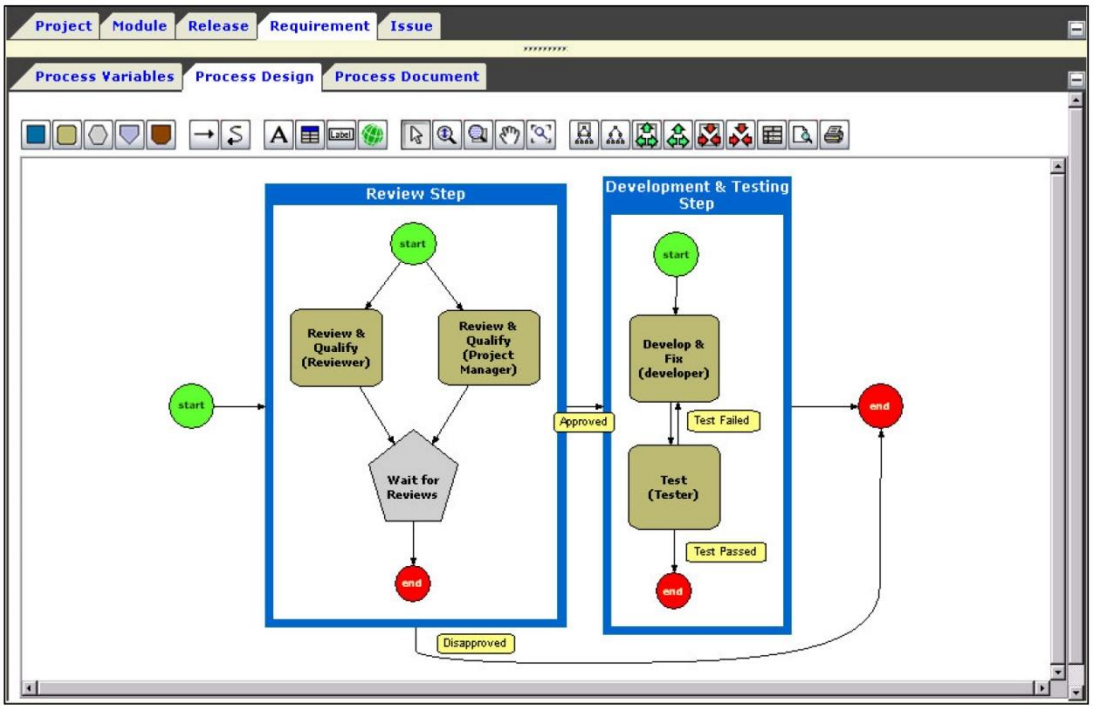


Fig: Kovair Visual Process Designer showing the Sample Lifecycle Process

About Kovair

Kovair Software is a Silicon Valley based software product company specializing in the domain of Integrated Application Lifecycle Management (ALM) solutions and supports global software development and management. Kovair's focus on integrating third party best-of-breed ALM tools enables creation of applications in a synchronized tools environment.

Kovair has partnered with leading technology brands like Microsoft, IBM, CA, BMC and more to provide customers a wide range of integration solutions.

Product Portfolio: Kovair's flagship products Omnibus Integration Platform, ALM Studio, QuickSync and Integrated DevOps are highly preferred solutions by some of the major corporations globally.

Recognitions: The SD Times 100 has recognized Kovair as one of the top 100 software innovators in the domain of Application Lifecycle Management. Kovair's Innovations in ALM Tools and ALM Integrations are well recognized both in the industry and by analysts at places like Gartner and Forrester.

Business Focus: Application Lifecycle Management Products and Services, Integration Platform

Industry Verticals: IT Consulting and Services, Banking and Financial Services, Telecom, Manufacturing, Networking, Healthcare, Defense and Government.

Contact: For more information about product and services contact sales@kovair.com. You may follow Kovair updates on Facebook, LinkedIn, Twitter, Google+, Slideshare and YouTube.

Important Links: Why Kovair| Management | Product Updates | Tool Integrations | Product Brochure | Videos | Datasheets | White Papers | Case Study | Technical Documents| Presentations | Services | Blog | Press Releases | Events | Customers | Partners | Support | Contact | Site Map

Global Technology Partners

Memberships and Associations