

# Mind-Alliance Test Plan

	Author(s)	Priyanka Vijay Gurav
	Reviewer(s)	Quamar Mehmood Siddiqui
	Dev Owner(s)	Jean-Francois Cloutier

## @ Document Revision History

Date	Revision	Who	Version	Remark
May 6,2011	Initial Draft	Quamar Mehmood	1.0	Initial draft
Sep 10,2011	Second Draft	Priyanka Gurav	2.0	Modified as per new features in Channels

## Table of contents

### Contents

<b>1. Feature overview and Scope</b>	<b>3</b>
1.1 Assumptions	3
1.2 Test Scope	3
<b>2. Build Acceptance Criteria &amp; Test Release Criteria</b>	<b>4</b>
2.1 Build Acceptance Criteria	4
2.2 Release Criteria	4
<b>3. Core Test Strategy</b>	<b>4</b>
3.1 Functionality	4
3.2 Unit Tests	15
3.3 Error/ Exception Handling	15
3.4 Browser Compatibility Testing	16
3.5 UI / UX Testing	17
3.6 Performance Testing / Concurrent User Test	19
3.7 Security Testing	20
3.8 Build Upgrading	20
<b>4. Communication and Reporting</b>	<b>21</b>
<b>5. Roles and Responsibilities</b>	<b>21</b>
<b>6. Defect Management</b>	<b>22</b>
<b>7. Test Case Management</b>	<b>22</b>
<b>8. Information and Knowledge Management</b>	<b>23</b>
<b>9. Test Automation</b>	<b>23</b>
<b>10. About AFour Technologies</b>	<b>24</b>
<b>11. AFour's Differentiators</b>	<b>25</b>

## 1. Feature overview and Scope

Mind-Alliance Systems, LLC is a Roseland, New Jersey-based software and management consulting firm that helps public and private sector clients share information and manage knowledge more effectively so they can collaboratively manage risk.

Channels provides planners, consultants, and enterprise architects with the most efficient framework available for creating information sharing plans and communication procedures for homeland security, emergency preparedness, business continuity, and operational risk management. These procedures detail what information needs to be sent and received so multiple organizations or business units can operate jointly.

Current document describes the test approach and test coverage for Channels functionality.

### 1.1 Assumptions

- It is assumed that the application is stable and updated on regular interval.
- Show stopper issues will be fixed immediately by dev team.
- Dev team will provide release notes along with installation instructions.

### 1.2 Test Scope

1. Test approach to channels testing
2. Test process elements such as test cycle, release criteria
3. Functional testing of channel administrator features
4. Functional testing of plan and segment bar
5. Functional testing of task and flow panel
6. Functional testing of information flow map
7. Functional testing of all social panels and collaboration panel
8. Functional testing of channels commands
9. Functional testing of display control
10. Functional testing of entities (e.g. Profiling and organization, Defining and event etc.) of channels
11. Strategy of functional testing and nonfunctional testing (performance optimization, compatibility)
12. Test automation of automatable test cases of above test areas
13. Locking behavior of components for multi user access
14. Functional test of Advance form and Simple form under task and flow panel. (New feature).
15. Functional testing of Information needs of all Participants features.
16. Functional testing of Is Guidelines for all Participants features.
17. Regression testing for Channels.

## 2. Build Acceptance Criteria & Test Release Criteria

### 2.1 Build Acceptance Criteria

1. Release notification is available with the build
2. All unit test issues must be fixed before BVT execution. BVT testing is performed by test team on every build
3. Test cases for BVT are selected from existing test case repository
4. The build must be installed successfully and all the BVT test cases must pass before starting test pass execution

### 2.2 Release Criteria

1. All L0 and L1 test cases are passing
2. All major+ priority bugs are closed
3. All modules are at ZBB (Zero Bug Bounce) for the agreed features in the release

## 3. Core Test Strategy

Test cases are categorized as following:

1. L0 – BVT test cases
2. L1 – Other functional positive test case variations
3. L2 – Negative test cases and nonfunctional test cases
4. L3 – Boundary test cases
5. L4 – Other test cases (e.g. UI / UX, Browser Compatibility, etc.)

### 3.1 Functionality

#	Test Area	Scenarios	Verifications
1	Channel Software Installation / Upgrade / Un-installation	<ul style="list-style-type: none"> <li>• Install Channel on new machine, downloaded from SVN.</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that War file created successfully after running maven command.</li> <li>• Verify Channel Software is installed successfully.</li> <li>• Verify Channel installed without an error.</li> </ul>
2		<ul style="list-style-type: none"> <li>• Upgrade Channel software.</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that Channel gets upgraded.</li> <li>• Verify that data should not be lost.</li> </ul>
3		<ul style="list-style-type: none"> <li>• Un-install Channel</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that un-installation removes all the</li> </ul>

		software.	components of Channel.
4	Channel Login.	<ul style="list-style-type: none"> <li>Successful Channel login</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user is able to access Channel URL.</li> <li>Verify user can login with valid credentials</li> </ul>
5		<ul style="list-style-type: none"> <li>Unsuccessful Channel Login.</li> </ul>	<ul style="list-style-type: none"> <li>Verify user cannot login with invalid credential.</li> </ul>
6	Channel Admin Management	<ul style="list-style-type: none"> <li>Create Plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that admin is able to create the new plan.</li> </ul>
7		<ul style="list-style-type: none"> <li>Delete Plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that admin is able to delete the plan.</li> </ul>
8		<ul style="list-style-type: none"> <li>Productize Plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that plan can be productize.</li> </ul>
9		<ul style="list-style-type: none"> <li>Update Plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that newly added plan should get updated</li> </ul>
10		<ul style="list-style-type: none"> <li>Create User</li> </ul>	<ul style="list-style-type: none"> <li>Verify that admin is able to create user.</li> </ul>
11		<ul style="list-style-type: none"> <li>Delete User</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user should be get deleted.</li> </ul>
12		<ul style="list-style-type: none"> <li>Assign Roles</li> </ul>	<ul style="list-style-type: none"> <li>Verify that Roles can be assigned</li> </ul>
13		<ul style="list-style-type: none"> <li>Disable User</li> </ul>	<ul style="list-style-type: none"> <li>Verify that admin can make user as disable.</li> <li>Verify that disable user should not able to log in.</li> </ul>
14		<ul style="list-style-type: none"> <li>Assign user to plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user is able to view the plan</li> </ul>
15		<ul style="list-style-type: none"> <li>Change role from Admin to Planner</li> </ul>	<ul style="list-style-type: none"> <li>Verify that admin is able to change the user role.</li> <li>Verify that User cannot view Channel Administration Link</li> </ul>
16		<ul style="list-style-type: none"> <li>Assign users to all plan</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user should be assign to the entire plan.</li> <li>Verify that user is able to view all plans</li> </ul>
17		<ul style="list-style-type: none"> <li>Create Plan with same name</li> </ul>	<ul style="list-style-type: none"> <li>Verify that plan should not be created with same name</li> <li>Valid error message should be generated</li> </ul>
18		<ul style="list-style-type: none"> <li>Create Plan with special Character</li> </ul>	<ul style="list-style-type: none"> <li>Verify that plan should not be created with special characters</li> <li>Valid error message should be generated</li> </ul>
19		<ul style="list-style-type: none"> <li>Create User with</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user should not be created with same name</li> </ul>

		same name	<ul style="list-style-type: none"> <li>Valid error message should be generated</li> </ul>
20		<ul style="list-style-type: none"> <li>Create user with special Character</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user should not be created with special characters</li> <li>Valid error message should be generated</li> </ul>
21		<ul style="list-style-type: none"> <li>Create user with invalid email id</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user with invalid email id should not be created.</li> <li>Valid error message should be generated</li> </ul>
22		<ul style="list-style-type: none"> <li>Enter valid email id in send planned feedback to</li> </ul>	<ul style="list-style-type: none"> <li>Verify that feedback should be send</li> </ul>
23		<ul style="list-style-type: none"> <li>Enter invalid email id in send planner feedback</li> </ul>	<ul style="list-style-type: none"> <li>Valid error message should be generated.</li> <li>Verify that feedback should not be send</li> </ul>
24		<ul style="list-style-type: none"> <li>Create Plan name without Owner</li> </ul>	<ul style="list-style-type: none"> <li>Verify that Plan should get created</li> </ul>
25		<ul style="list-style-type: none"> <li>Update Owner name</li> </ul>	<ul style="list-style-type: none"> <li>Verify that owner name is modified</li> </ul>
26		<ul style="list-style-type: none"> <li>Delete owner name</li> </ul>	<ul style="list-style-type: none"> <li>Verify that owner name is blank</li> </ul>
		<ul style="list-style-type: none"> <li>Navigate to Home page</li> </ul>	<ul style="list-style-type: none"> <li>Verify that user get navigate to Home page after clicking Home link.</li> </ul>
27	Home Page Management	<ul style="list-style-type: none"> <li>Click on Channel Administration Link</li> </ul>	<ul style="list-style-type: none"> <li>Verify that page with Channel Administration opens</li> </ul>
28		<ul style="list-style-type: none"> <li>Click on Information Sharing model Link</li> </ul>	<ul style="list-style-type: none"> <li>Verify that page with Channel :&lt;Plan name&gt; opens</li> </ul>
29		<ul style="list-style-type: none"> <li>Click on Procedure Explorer Link</li> </ul>	<ul style="list-style-type: none"> <li>Verify that page with All Procedures in &lt;Plan name&gt; opens</li> </ul>
30		<ul style="list-style-type: none"> <li>Click on Procedures for all Participants Link</li> </ul>	<ul style="list-style-type: none"> <li>Verify that page with Responder with &lt;Plan name&gt; opens</li> </ul>

31		• Send Feedback	• Verify that feedback can be sent to the valid email address
32		• Send Message	• Verify that message can be send
33		• Send message to everyone	• Verify that message can be send to everyone
34		• Send message to All planners	• Verify that message can be send to all planners
35		• Delete Sent Messages	• Verify that sent messages can be deleted
36		• Email sent message	• Verify that message can be emailed
37		• Resend Email	• Verify that email can be resend
38		• Message received	• Verify that message can be received
39		• Survey -Hide completed	• Verify that survey should be hidden
40		• Survey – show all	• Verify that survey should be shown
41		• Calendar	• Verify that calendar can be seen in Channels with details
42		• All Events	• Verify that all events can be seen in calendar
43		• Print Calendar details	• Verify that calendar details can be printed
44		• Edit Calendar details	• Verify that Calendar can be edited
45		• Today date	• Verify that Today's date and details can be seen
46		• Learn More	• Verify that Google calendar is opened.
47	Display Controls	• Show flow map legend	• Verify that Flow Diagram legend can be seen

48		<ul style="list-style-type: none"> <li>Stretch up form</li> </ul>	<ul style="list-style-type: none"> <li>Verify that form can be stretch up.</li> <li>Verify that form can be shrink back</li> </ul>
49		<ul style="list-style-type: none"> <li>Reduce flow map to fit</li> </ul>	<ul style="list-style-type: none"> <li>Verify that flow map is reduced to fit within the available space</li> <li>Verify clicking on the icon once more brings the flow map back to its normal size.</li> </ul>
50		<ul style="list-style-type: none"> <li>Maximize flow map</li> </ul>	<ul style="list-style-type: none"> <li>Verify that flow map can be maximize to use all of the screen space allocated to plan editor</li> </ul>
51		<ul style="list-style-type: none"> <li>Minimize flow map</li> </ul>	<ul style="list-style-type: none"> <li>Verify that flow map can be minimized.</li> <li>Verify that flow map also minimize by selecting task or flow within the map</li> </ul>
52		<ul style="list-style-type: none"> <li>Show / Hide Goals</li> </ul>	<ul style="list-style-type: none"> <li>Verify that Goals (gains) and risks are displayed on flow map</li> <li>Verify clicking again icon hides the goals.</li> </ul>
53		<ul style="list-style-type: none"> <li>Show / Hide Info needs and capabilities</li> </ul>	<ul style="list-style-type: none"> <li>Verify that when planner clicks on icon it displays info needs and capabilities (tasks connected by arrows to square dots).</li> <li>Verify that clicking again icon hides the info needs and capabilities.</li> </ul>
54		<ul style="list-style-type: none"> <li>Hide / Show non-operational task and flow</li> </ul>	<ul style="list-style-type: none"> <li>Verify that non-operation task and flow can be hidden and shown</li> </ul>
55	Channel Command Execution	<ul style="list-style-type: none"> <li>Verify Undo and Redo command operation.</li> </ul>	<ul style="list-style-type: none"> <li>Verify Do / Undo / Redo operation for following commands.               <ol style="list-style-type: none"> <li>Add Segment (Update)</li> <li>Remove Segment</li> <li>Add Goal</li> <li>Remove Goal</li> <li>Add New Task</li> <li>Cut Task</li> <li>Paste Task Using Cut</li> <li>Paste Task Using Copy</li> <li>Duplicate Task</li> <li>Intermediate Task</li> <li>Disintermediate Task</li> <li>Move Task</li> <li>Connect Flow</li> <li>Update Set</li> <li>Add Issue</li> </ol> </li> </ul>



			<ul style="list-style-type: none"> <li>16. Remove Issue</li> <li>17. Attach Document</li> <li>18. Copy Attachment</li> <li>19. Cut Attachment</li> <li>20. Paste Attachment</li> <li>21. Copy Task</li> <li>22. Set Task From Copy</li> <li>23. Add Info Need</li> <li>24. Add Info Capability</li> <li>25. Add Flow</li> <li>26. Duplicate Flow</li> <li>27. Break Up Flow</li> <li>28. Copy Flow</li> <li>29. Transfer Job</li> <li>30. Create Entity</li> <li>31. Update Add</li> <li>32. Update Move</li> <li>33. Update Remove</li> </ul>
56	Collaboration Panel	<ul style="list-style-type: none"> <li>• Hide collaboration Panel</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that panel can be hidden</li> </ul>
57		<ul style="list-style-type: none"> <li>• Show collaboration panel (Planner list)</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that collaboration panel can be seen</li> </ul>
58		<ul style="list-style-type: none"> <li>• Presence tab</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that list shows all the logged in users</li> </ul>
59		<ul style="list-style-type: none"> <li>• Show / Hide inactive users</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that list shows all users (logged in as well as logged off) when user click on "Show all user" link</li> <li>• Verify that only logged in users are visible when user select "Hide inactive user"</li> </ul>
60		<ul style="list-style-type: none"> <li>• Send Message to inactive user from Presence tab</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be sent to inactive user</li> </ul>
61		<ul style="list-style-type: none"> <li>• Activities</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that list of activities done by all the logged in users</li> </ul>
62		<ul style="list-style-type: none"> <li>• Hide my activities</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that all activities performed by active user can be hidden</li> </ul>
63		<ul style="list-style-type: none"> <li>• Show all activities</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that all activities of all users can be seen</li> </ul>

64		<ul style="list-style-type: none"> <li>• Show more activities</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that list shows all activities performed by user</li> </ul>
65		<ul style="list-style-type: none"> <li>• Show few activities</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that activities can be fit to page and shows recent activities</li> </ul>
66		<ul style="list-style-type: none"> <li>• Send Message to user from Activities tab</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that a message can be send</li> </ul>
67		<ul style="list-style-type: none"> <li>• Send message to selected user from Message tab</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send to selected user</li> <li>• Verify that the recipient received the message</li> </ul>
68		<ul style="list-style-type: none"> <li>• Send &amp; Email message to selected user from message tab</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send and emailed to selected user</li> <li>• Verify that recipient received the message and email</li> </ul>
69		<ul style="list-style-type: none"> <li>• Send message to all planners</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send to all planners</li> <li>• Verify that all planners should receive the message</li> </ul>
70		<ul style="list-style-type: none"> <li>• Send &amp; Email message to all planner</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send and emailed to all planners</li> <li>• Verify that all planners should receive the message and email</li> </ul>
71		<ul style="list-style-type: none"> <li>• Send message to everyone</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send to everyone i.e. (To admin/ lanner/User)</li> <li>• Verify that everyone should receive the message</li> </ul>
72		<ul style="list-style-type: none"> <li>• Send &amp; Email message to everyone</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be send and emailed to every</li> <li>• Verify that everyone received the message</li> </ul>
73		<ul style="list-style-type: none"> <li>• Hide broadcasts (sent messages)</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that the broadcast message can be hidden</li> </ul>
74		<ul style="list-style-type: none"> <li>• Show all messages</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that all the message are shown</li> </ul>
75		<ul style="list-style-type: none"> <li>• Delete this message (Selected message)</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that message can be deleted</li> <li>• Verify that message can also be deleted from the recipients screen</li> </ul>
76		<ul style="list-style-type: none"> <li>• Show received</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that received message can be seen</li> </ul>

		messages	
77		• Show sent messages	• Verify that sent messages can be seen
78		• Reset	• Verify that message text box reset to blank
79		• Email this message (Selected message)	• Verify that message can be emailed
80		• Resend email	• Verify that message can be emailed
81	Plan and Segment Bar	• Show menu	• Verify that show menu give access to collaboration panel if collapsed, about plan, about plan segment, survey, procedure report, mapped procedure, all segments, all issues, all types, index, help
82		• Action menu	• Verify that action menu enables the planner to act on the segment as a whole by-sending a message, paste task, attachment ,add issue, segment, move segment, remove segment and logout user.
83		• Help icon	• Verify that help page opens
84		• Back and Forward button	• Verify that planner can back or forward opened window
85		• Plan Segment link on plan page	• Verify that the about plan segment window opens • Verify that details can be added
86		• About plan	• Verify that window about plan opens • Verify that about plan details can be modified • Verify that About plan show menu gives access to details, all events, all organizations ,all types, all issues, index, participants, versions , who's who, bibliography, all tags, evaluations, procedure map
87		• About plan action	• Verify that action menu enables to send message and add new issue, undo, redo, paste
88		• About plan segment	• Verify that window about plan segment opens • Verify that plan segment can be modified • Verify that show gives access to details, goal, organizations, map, task mover
89		• About plan segment	• Verify that action menu enables to send message and add new issue, undo, redo, paste

		action	
90		• Plan Events	<ul style="list-style-type: none"> <li>• Verify that new events can be added</li> <li>• Verify that details for event can be added</li> </ul>
91		• Plan Organizations	<ul style="list-style-type: none"> <li>• Verify that new organization can be added</li> <li>• Verify that details for organization can be added</li> </ul>
92		• Plan Procedure Map, All Segment, Who's Who, All Issue, Bibliography, All Index, All Types, All Tags, Evaluation, Participation, Versions, Action	<ul style="list-style-type: none"> <li>• Verify that planner is able to access mentioned components</li> </ul>
93		• Plan Survey	<ul style="list-style-type: none"> <li>• Verify that survey window opens</li> <li>• Verify that planner is able to access plan survey</li> </ul>
94		• Procedure Report	<ul style="list-style-type: none"> <li>• Verify that Procedure-anyone from organization opens</li> <li>• Verify that planner is able to access Procedure report</li> </ul>
95		• Mapped Procedure	<ul style="list-style-type: none"> <li>• Verify that All procedure in &lt;plan name&gt;</li> <li>• Verify that planner is able to access Mapped Procedure</li> </ul>
96		• Help form about plan	<ul style="list-style-type: none"> <li>• Verify that help page opens</li> </ul>
97		• Move task to segment	<ul style="list-style-type: none"> <li>• Verify that window with task mover opens</li> <li>• Verify that planner can move task from one segment to other segment</li> </ul>
99		• Add new issue	<ul style="list-style-type: none"> <li>• Verify that issue can be added</li> </ul>
100		• Add new segment to plan	<ul style="list-style-type: none"> <li>• Verify that segment can be added</li> <li>• Verify that details can be entered</li> <li>• Verify that about plan segment have access to goals, details, organizations, map, task mover</li> </ul>
101		• Add new task to segment	<ul style="list-style-type: none"> <li>• Verify that task can be added</li> <li>• Verify that details of task can be entered</li> </ul>
102		• Remove Segment	<ul style="list-style-type: none"> <li>• Verify that segment can be removed</li> </ul>

		from plan	
103		• Logout from plan page	• Verify that user can logout from the channels
104	Task and Flow Panel	• Receives panel	• Verify that Receives panel is present
105		• Sends panel	• Verify that Send panel is present
106		• Add info receives	<ul style="list-style-type: none"> <li>• Verify that info can be received</li> <li>• Verify that details can be entered</li> <li>• Verify that details can be modified</li> </ul>
107		• Add info sent	<ul style="list-style-type: none"> <li>• Verify that info can be send</li> <li>• Verify that details can be entered</li> <li>• Verify that details can be modified</li> </ul>
108		• Create Flow	<ul style="list-style-type: none"> <li>• Verify that flow can be created</li> <li>• Verify that flow can be modified</li> </ul>
109		• Expand task summery	• Verify that summaries can be expanded in form
110		• Color of bullet	• Verify that the color of bullet signified the how important the task is
111	Entities	• Verify CRUD (Create / Read / Update and Delete) operation on entities of plan	<ul style="list-style-type: none"> <li>• Verify CRUD operation for following Entities <ul style="list-style-type: none"> <li>1. Agent</li> <li>2. Places</li> <li>3. Transmission media</li> <li>4. Event</li> <li>5. Phase</li> <li>6. Organizations</li> <li>7. Network entity</li> <li>8. Issue View</li> </ul> </li> </ul>
112	Information Flow map	• Start of segment event phase	• Verify that event is green
113		• End of segment	• Verify that event is red
114		• Information	• Verify that information can be seen on the arrow
115		• Intent-Only if task	• Verify that flow is dotted

		fails	
116		<ul style="list-style-type: none"> <li>• Sender's task-Terminates</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that flow contains stop information</li> </ul>
117	Procedure for all participants	<ul style="list-style-type: none"> <li>• Procedure for all participant page</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that page with Responder list for &lt;plan name&gt;</li> </ul>
118		<ul style="list-style-type: none"> <li>• Other planners</li> </ul>	<ul style="list-style-type: none"> <li>• Verify list of all planners who can edit this plan</li> <li>• Verify that email can be sent to the list of user from the outlook to the specified users</li> </ul>
119		<ul style="list-style-type: none"> <li>• Other plans</li> </ul>	<ul style="list-style-type: none"> <li>• Verify list of all the plans which logged user can edit</li> <li>• Verify that when click on &lt;Plan name&gt; the name of agent comes if available</li> <li>• Verify that when double clicked the agents get disabled</li> </ul>
120		<ul style="list-style-type: none"> <li>• Agent</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that list of task assigned to the agent</li> <li>• Verify that person defined in plan can be assigned to users</li> <li>• Verify that person can remove the assigned user</li> </ul>
121	Lock functionality	<ul style="list-style-type: none"> <li>• Lock on Channel components</li> </ul>	<ul style="list-style-type: none"> <li>• Verify lock on following Channel components when multiple user accessing the same component               <ol style="list-style-type: none"> <li>About Plan Details</li> <li>About Plan Events</li> <li>About Plan Organization</li> <li>About Plan Segment Details</li> <li>About Plan Segment Goal</li> <li>Update Task</li> <li>Update Flow</li> <li>Update Phase</li> <li>Update Agent</li> <li>Update Place</li> <li>Update Transmission Media</li> <li>Update Organization</li> <li>Update Role</li> <li>Move Task</li> </ol> </li> </ul>
122	Advance Form and Simple Form of Task	<ul style="list-style-type: none"> <li>• Advance Form and Simple Form of Task</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that Advance form gets open when clicked on advance form link of task.</li> <li>• Verify that Simple form gets open when clicked on</li> </ul>

	and Receive and Send info.		simple form of task.
123		<ul style="list-style-type: none"> <li>• Advance Form and Simple Form of Receive and Send info.</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that Advance form of receive info or send info gets open when clicked on advance form link of receive or send info.</li> <li>• Verify that Simple form of receive info or send info gets open when clicked in simple form link of receive or send info.</li> </ul>
125	Conceptual task or flow	<ul style="list-style-type: none"> <li>• Task is Conceptual</li> </ul>	<ul style="list-style-type: none"> <li>• Verify if task is conceptual and agent is assigned to the task</li> <li>• it has been declared so by a planner,</li> <li>• or it has no assignment,</li> <li>• or if all assignments are conceptual.</li> </ul>
		<ul style="list-style-type: none"> <li>• Flow is Conceptual</li> </ul>	<ul style="list-style-type: none"> <li>• Verify if flow is conceptual and</li> <li>• It has been declared so by a planner</li> <li>• or either source or target task is conceptual,</li> <li>• or no communication channel is identified,</li> <li>• or no information sharing commitment can be derived from it,</li> <li>• or if all commitments derived are conceptual.</li> </ul>
124	Regression Testing	<ul style="list-style-type: none"> <li>• Task Panel</li> </ul>	<ul style="list-style-type: none"> <li>• Verify that if created task with agent, organization, events and goals gets created.</li> </ul>

### 3.2 Unit Tests

TBD: Need to review code for preparing unit test cases.




### 3.3 Error/ Exception Handling

We have capture stack trace during automation in a text file also we have issue defect in trac. We have capture the error message e.g. Error message on Channel Administration page for duplicate plan like plan with same URI already exist. And have written test cases for such error message.

Sr. no.	Component	Description	Error Message
1	Channel Login Page	Recover password using Invalid "User id"	Unrecognized user name
2	Channel Home Page	Feedback send to invalid email or email that does not exists	Enter Valid Email Address
3		Sent Message as Blank Spaces	Message is empty. Not sent

4		Sent & Email Message as Blank Spaces	Message is empty. Not sent
5		Name is removed under About me tab of social panel	The name is required
6		Name is removed under About me tab of social panel	No changes were made
7		Email is removed under About me tab of social panel.	An email id is requires
8		Email is removed under About me tab of social panel.	No changes were made
9	Channel Administration Page	Plan with same URI	A plan with that uri already exists
10		Same user name	User <user> already exists
11		Invalid email id	'abc@.com' is not a valid email address.
12	Channel Plan Page	Sent Message as Blank Spaces under Message tab of collaboration panel.	Message is empty. Not sent
13		Sent & Email Message as Blank Spaces under Message tab of collaboration panel.	Message is empty. Not sent
14		Send blank feedback under help	Please enter short text.

### 3.4 Browser Compatibility Testing

OS / Browser	Windows 7	Unix Platform (Ubuntu Server)
 Internet Explorer 8	✓	✓
 Firefox 3.6	✓	✓
 Google Chrome	✓	✓



#### LEGEND

Symbol	Description
✓	Fully compatible
P	Partially compatible (some features may not work)
N	Not Tested
X	Not compatible

### 3.5 UI / UX Testing

#	Test Area	Description
1	All Views (Channels web pages)	Ensure that all the views are covered and verified. Refer Mind-AllianceTestCaseSheet.ods or Mind-AllianceViewsCoverageSheet.ods.
2	Target page	Ensure that maximum 3 mouse click should be performed to access any page
3	Required Fields	If the screen requires data entry on a specific field, designers should identify the required fields with a red asterisk and generate a friendly warning if the data is left blank.
4	Data Type Errors	If the screen contains dates, numeric, currency or other specific data types, ensure that only valid data can be entered.
5	Field Widths	If the screen contains text boxes that allow data entry, ensure that the width of data entered does not exceed the width of the table field (e.g. a title that is limited to 100 characters in the database should not allow more than 100 characters to be entered from the user interface).
6	Onscreen Instructions	Any screen that is not self-explanatory to the casual user should contain onscreen instructions that aid the user.
7	Keep Onscreen Instructions Brief	While onscreen instructions are great, keep the wording informative, in layman's terms, but concise.
8	Progress Bars	If the screen takes more than 5 seconds to render results, it should contain a progress bar so that the user understands the processing is continuing.
9	Same Document Opened Multiple Times	If the application opens the same document multiple times, it should append a unique number to the open document to keep one document from overwriting another. For example, if the application opens a document named Minutes.txt and it opens the same document for the same user again, consider having it append the time to the document or sequentially

		number it (Minutes2.txt or Minutes_032321.txt).
10	Cosmetic Inconsistencies	The screen look, feel, and design should match the other screens in your application. Creating and using a style guide is a great way to ensure consistency throughout your application.
11	Abbreviation Inconsistencies	If the screens contain abbreviations (e.g. No for number, Amt for amount, etc), the abbreviations should be consistent for all screens in your application. Again, the style guide is key for ensuring this.
12	Save Confirmations	If the screen allows changing of data without saving, it should prompt users to save if they move to another record or screen.
13	Delete Confirmations	If a person deletes an item, it is a good idea to confirm the delete. However, if the user interface allows deleting several records in a row, in some cases developers should consider allowing them to ignore the confirmation as it might get frustrating to click the confirmation over and over again.
14	Type Ahead	If the user interface uses combo boxes (drop down lists), be sure to include type ahead (if there are hundreds of items in a list, users should be able to skip to the first item that begins with that letter when they type in the first letter).
15	Grammar and Spelling	Ensure the test cases look for grammar or spelling errors.
16	Table Scrolling	If the application lists information in table format and the data in the table extend past one page, the scrolling should scroll the data but leave the table headers intact.
17	Error Logging	If fatal errors occur as users use your application, ensure that the application writes those errors to a log file, event viewer, or a database table for later review. Log the routine the error was in, the person logged on, and the date/time of the error.
18	Error Messages	Ensure that error messages are informative, grammatically correct, and not condescending.
19	Shortcuts	If the application allows short cut keys (like CTRL+S to save), test each shortcut to ensure it works in all different browsers (if the application is web based).
20	Invalid Choices	Do not include instructions for choices not available at the time. For example, if a screen cannot be printed due to the state of the data, the screen should not have a Print button.
21	Invalid Menu Items	Do not show menu items that are not available for the context users are currently in.
22	Dialog Box Consistency	Use a style guide to document what choices are available for dialog boxes. Designers should not have Save/Cancel dialog on one screen and an OK/Cancel on another. This is inconsistent.
23	Screen Font Type	Ensure that the screen font family matches from screen to screen. Mismatching fonts within the same sentence and overuse of different fonts

		can detract from the professionalism of your software user interface.
24	Screen Font Sizes	Ensure that the screen font sizes match from screen to screen. A good user interface will have an accompanying style guide that explicitly defines the font type and size for headers, body text, footers, etc.
25	Colors	Ensure that screens do not use different color sets as to cause an inconsistent and poorly thought-out user interface design. Your style guide should define header colors, body background colors, footer colors, etc.
26	Icons	Ensure that icons are consistent throughout your application by using a common icon set. For example, a BACK link that contains an icon next to it should not have a different icon on one screen versus another. Avoid free clip-art icons; opt for professionally designed icons that complement the overall look and feel of your screen design.
27	Narrative Text	Having narrative text (screen instructions) is a great way to communicate how to use a specific screen. Ensure that narrative text appears at the same location on the screen on all screens.
28	Brevity	Ensure that narrative text, error messages and other instructions are presented in laymen's terms but are brief and to-the-point.
29	Dialog Box Consistency	Use a style guide to document what choices are available for dialog boxes. You should not have Save/Cancel dialog on one screen and an OK/Cancel on another, this is inconsistent.
30	Links	If your application has links on the screen (e.g. Save as Spreadsheet, Export, Print, Email, etc.), ensure that the links have consistent spacing between them and other links, that the links appear in the same order from screen to screen, and that the color of the links are consistent.
31	Menus	If your application has menu items, ensure that menu items that are not applicable for the specific screen are disabled and the order in which each menu item appears is consistent from screen to screen.
32	Buttons	If your application has buttons (e.g. Submit, OK, Cancel, etc.), ensure that the buttons appear in a consistent order from screen to screen (e.g. Submit then Cancel).

### 3.6 Performance Testing / Concurrent User Test

Performance test / concurrent user testing is performed to measure / observe the Channel behavior under working load. Jmeter tool is used for developing the scripts and conducting test. Test is conducted with number of threads with mix of transaction as per the required percentage.

Test was conducted to make sure that following high level feature should not break:

- a. Lock behavior. I.e. Multiple planners should not modify the same components at the same time. If multiple planner are accessing the channels components (e.g.

segment, task) then a lock should be generated on these components and only one planner can access the channel components.

- b. Multiple planners should be able to access different components of the channels. E.g. If planner A is updating plan details other planner can add segments to the plan.
- c. Multiple planner should access different components and perform activities on Advance features like transfer jobs, create entity with attachment, show if task is conceptual.

## Test Matrix

Business Metrics	Software Metrics
<ul style="list-style-type: none"> <li>Standard dev should be less than 2 secs.</li> <li>Response time should be for per page not more than 5 secs.</li> </ul>	<ul style="list-style-type: none"> <li>CPU utilization should be less than 70%</li> <li><b>Avg. Disk Queue Length:</b> should not exceed exceeds a value of 2 during peak usage</li> <li><b>Avg. Disk Sec/Read:</b> <ul style="list-style-type: none"> <li>Less than 10 ms - very good</li> <li>Between 10 - 20 ms - okay</li> <li>Between 20 - 50 ms - slow, needs attention</li> <li>Greater than 50 ms – Serious I/O bottleneck</li> </ul> </li> <li><b>Avg. Disk Reads/Sec:</b> should be less than 85 percent of the disk capacity</li> <li><b>Physical Disk: %Disk Time:</b> if this value is greater than 50 percent, it represents an I/O bottleneck.</li> <li><b>Avg. Disk Writes/Sec:</b> should be less than 85 percent of the disk capacity</li> </ul>

Refer “**MindAllianceConcurrentTestcaseSheet.ods**” file for scenarios.

## 3.7 Security Testing

- Exposed interfaces of the system will be identified
- Threat modeling of each interface will be done using STRIDE Threat Model
- Report of security and privacy related vulnerabilities will be identified
- Controlled attacks will be done to verify system security against the identified vulnerabilities

## 3.8 Build Upgrading.

Test Area	Key Verification
Upgrade major releases	<ul style="list-style-type: none"> <li>Verify that Channel gets upgraded.</li> <li>Verify that core functionality is intact</li> <li>Verify that one can access old plan and retrieve all the components (Users / Segments / Tasks / Organizations, etc. ) and the data associated with the components (No data loss)</li> </ul>
Upgrade minor releases	<ul style="list-style-type: none"> <li>Verify that Channel gets upgraded.</li> <li>Verify that core functionality is intact</li> <li>Verify that one can access old plan and retrieve all the components (Users / Segments / Tasks / Organizations, etc. ) and the data associated with the components (No data loss)</li> </ul>

#### 4. Communication and Reporting

Document Type	What For	Who Sends and When
Daily Status Report	Daily Activity Status and Issues / Queries / Concerns	SDET 1 daily at the end of day (IST)
Weekly Status Report	Goals Achieved in Current Week Goals Planned for Next Week Issues / Queries / Concerns Reference to Test Status Report	SDET 1 at the end of every week.
Monthly Status Report	Goals achieved in current month Goals Planned for Next Month Issues / Queries / Concerns	SDET 1 at the end of month.
Test Status Report	Top 15 Defects Test Pass Statistics Defect Status Automated Test Execution Statistics	SDET 1

#### 5. Roles and Responsibilities

Role	Responsibilities
SDET 1	Test Planning Test Design Test Elaboration and Test Execution Defect Logging and Tracking Test Results Reporting

	Test Case Management Test Automation Performance Test Development Performance Test Analysis Communication through Calls / Demos and Emails Daily Status and Weekly Status Reporting
SDET 2	Test Design Test Elaboration and Test Execution Defect Logging and Tracking Test Results Reporting Test Case Management Test Automation Performance Test Development

## 6. Defect Management

1. All defects will be logged by test team in trac.
2. Bugs will be closed by test team members after verification of issues resolved by development team members
3. All defects logged will have a test case documented in test case repository. Few exceptions will be made where test case is not needed (Such as typo error or alignment bugs)

## 7. Test Case Management

All documents related to QA activities are stored in SVN repository with latest changes.  
Below is the list of documents:

- a. UI automation test case spread sheet.
- b. Concurrent user/ performance test spread sheet.
- c. Automation framework design document.
- d. Automation framework user guide document.
- e. Channels views test coverage sheet.
- f. Channels command test coverage sheet.
- g. Functional test case sheet.

A test case sheet should contain following sections.

- a. Test case id: Unique number or name. E.g. MAV0001\_ChannelLogin

- b. Test case name / Title: Test case title (feature to be tested)
- c. Priority: the importance of this test case w.r.t functionality
- d. Pre-condition necessary tasks to do before starting the test execution
- e. Test procedure / Steps: a step by step procedure to execute this test case
- f. Expected Result
- g. Actual Result
- h. Test case pass / fail criteria (Status): when this case is "pass", when this case is "fail"
- i. Comment

## 8. Information and Knowledge Management

Artifacts	Repository
Test Plan	<a href="http://wush.net/trac/mindalliance/browser/TestPlan">http://wush.net/trac/mindalliance/browser/TestPlan</a>
Test Case sheet	<a href="http://wush.net/trac/mindalliance/browser/TestCases">http://wush.net/trac/mindalliance/browser/TestCases</a>
Test Coverage sheet	<a href="http://wush.net/trac/mindalliance/browser/TestCoverage">http://wush.net/trac/mindalliance/browser/TestCoverage</a>
Defects logging and tracking	<a href="http://wush.net/trac/mindalliance/newticket">http://wush.net/trac/mindalliance/newticket</a>
Test automation scripts	<a href="http://wush.net/trac/mindalliance/browser/TestAutomation">http://wush.net/trac/mindalliance/browser/TestAutomation</a>
Performance test scripts	<a href="http://wush.net/trac/mindalliance/browser/PerformanceTests">http://wush.net/trac/mindalliance/browser/PerformanceTests</a>
Framework design and user guide	Documents will be placed under docs folder under respective repositories

## 9. Test Automation

Test Area	Approach	Tools and Technologies
UI Automation	<p>Test Cases of Views are targeted and automated to ensure sanity and robustness of UI.</p> <p>This provides insurance against breaking of parts of UI due to changes to the Channels.</p> <p>Test Coverage sheet is reviewed time to time and test cases are added and automated for new Views.</p> <p>Reference Documents:  Mind-AllianceTestAutomationFrameworkDesign.pdf  Mind-AllianceTestAutomationUserGuide.pdf</p>	Selenium with Web driver and core java for framework development through Eclipse IDE.
Functional Test	In phase 1 test case of Plan and Command Execution targeted for test automation	Selenium with Web driver and core java

Automation	<p>Test Coverage sheet is reviewed time to time and test cases are added and automated for changes to the plans and command execution functionality.</p> <p>In phase 2 functional test cases will be documented and automated in order of priority of test cases. This will serve as insurance against regression to the core functionality.</p> <p>Reference Documents: Mind-AllianceTestAutomationFrameworkDesign.pdf Mind-AllianceTestAutomationUserGuide.pdf</p>	for framework development through Eclipse IDE.
Performance Test Automation	<p>In phase 1, concurrent user tests, for up to 10 simultaneous users, were automated and executed for verification of component locking behavior.</p> <p>In phase 2, concurrent user tests, for up to 100+ simultaneous users were automated and executed for verification of system behavior under load.</p> <p>Iterations of above tests will be executed iteratively on every week to ensure that system does not introduce performance bottleneck or locking issues due to functional changes in the system.</p> <p>Reference Documents: MindAllianceConcurrentTestcaseSheet.ods</p>	Jmeter is used for developing scripts and conducting test.
API Test Automation	<p>List of key APIs will be created after code walk through and signed-off. API / Unit tests will be created for the key functions.</p> <p>These tests will be executed every week to ensure interface lock down and consistent behavior of API across builds.</p>	JUnit or Custom Test Harness, Core java framework development through Eclipse IDE.

## 10. About A4Four Technologies



AFour Technologies (AFour) is an Independent Testing Services organization with a singular focus on software testing and quality engineering. The emphasis on “Testing Only” as opposed to “Testing Also” has enabled us to develop deep expertise in all aspects of quality engineering, and has given us the required agility in meeting the business and quality objectives of our customers. AFour has offices in Pune, India, and in Bellevue, USA. AFour’s niche services include Test Strategy and Consulting, Test Automation, Performance, Security and Compatibility testing in various domains.

AFour has executed engagements for various companies from start-ups to large organizations and has a sustainable customer base. Some of the areas we have deep experience in are Test Automation, Performance, Security, Compatibility and Interoperability testing. We have expertise in all major 3<sup>rd</sup> party test automation tools such as QTP, SilkTest, RFT, ALM Platforms like VSTS, and Open source test automation tools and libraries including Selenium. In the mobility space, we have developed test automation frameworks for applications on handsets as well as on the server side, using tools like M-Eux, Robotium and iMacros. In addition, we have also developed custom tools and frameworks in mobility. We also have experience working with a range of Performance Testing tools like LoadRunner, JMeter, VSTS Load Test and AppPerfect LoadTest, and in Lab set up and test infrastructure management.

AFour has a strong test team of 30 expert test engineers, each with automation experience and specialized skills in testing. With our agility and nimbleness, we have been a trusted QA partner for several high-tech startups.

## **11. AFour’s Differentiators**

- “Expert Testing” corporate culture and mindset
- Test focused career paths enabling us to attract and retain top talent in testing
- Innovative training-based recruitment model
- Strong emphasis on Automation, Tool Development, and non-functional test attributes like Performance, Security, Reliability, Interoperability and Compatibility
- “A4Agile” Engagement Methodology
- Creative service models for test management infrastructure in the cloud