



# LANDESK Service Desk

---

Design Ideas

This document contains information, which is the confidential information and/or proprietary property of LANDESK Software, Inc. and its affiliates (referred to collectively as "LANDESK"), and may not be disclosed or copied without prior written consent of LANDESK.

To the maximum extent permitted under applicable law, LANDESK assumes no liability whatsoever, and disclaims any express or implied warranty, relating to the sale and/or use of LANDESK products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right, without limiting the rights under copyright.

LANDESK retains the right to make changes to this document or related product specifications and descriptions, at any time, without notice. LANDESK makes no warranty for the use of this document and assumes no responsibility for any errors that can appear in the document nor does it make a commitment to update the information contained herein. For the most current product information, please visit [www.landesk.com](http://www.landesk.com).

Copyright © 2011, 2014, LANDESK Software, Inc. and its affiliates. All rights reserved. LANDESK and its logos are registered trademarks or trademarks of LANDESK Software, Inc. and its affiliates in the United States and/or other countries. Other brands and names may be claimed as the property of others.

Document number LDSD\028 Issue 1.3

# Contents

<b>Contents</b> .....	<b>3</b>
<b>Design Ideas</b> .....	<b>5</b>
<b>General Design Considerations</b> .....	<b>6</b>
<b>Availability Management</b> .....	<b>7</b>
Cost fields .....	7
Reporting on Availability .....	7
<b>Capacity Management</b> .....	<b>8</b>
Tests and Risks .....	8
Planning .....	8
Simulation, Modelling and Forecasting .....	9
<b>Change Management</b> .....	<b>10</b>
Emergency, Normal and Standard Changes .....	10
Change Schedule, Forward Schedule of Change, and Projected Service Outages (PSOs) .....	10
Software Development and Enhancement .....	11
Change Type and Categorization .....	11
Approval and CAB .....	11
Backout plan .....	12
Change Review .....	12
<b>Event Management</b> .....	<b>13</b>
Using Event Manager for Comprehensive Event Management .....	13
Linking CIs to Services .....	13
Auditing Events .....	13
Notifications .....	13
Create a Current Events query .....	14
Automatic and manual actions .....	14
Automated recovery .....	14
Reporting on Event Management .....	14
Event Manager mappings .....	14
Handling different Event types .....	15
<b>Financial Management</b> .....	<b>17</b>
Financial Management Overview .....	17
Cost Allocation in Service Desk .....	17
Customer Charging .....	18
Budgeting .....	18
External Data .....	18
<b>Incident Management</b> .....	<b>19</b>
Creating other processes .....	19
<b>IT Service Continuity and Disaster Recovery</b> .....	<b>20</b>
Continuity overview .....	20
Test Planning .....	20
<b>Knowledge Management</b> .....	<b>22</b>
Numbering Articles .....	22
Knowledge relevance .....	22
Service-centric Knowledge .....	22
Validation and verification .....	22
External Links .....	23
Expiring Articles .....	23
Known Errors .....	23
Workarounds .....	23
Using Knowledge throughout the Service Lifecycle .....	24
Further uses for the Is Knowledgeable? property .....	24
Reporting on Knowledge .....	24

<b>Problem Management</b> .....	<b>25</b>
Impact, Urgency, Severity, Priority and Category .....	25
Known Errors .....	25
Other Problem processes .....	25
Major Problems .....	25
<b>Release Management</b> .....	<b>26</b>
Considering the bigger picture in Release Management .....	27
<b>Request Fulfilment and Service Requests</b> .....	<b>28</b>
Tool integration for automated software deployment .....	28
<b>Service Asset and Configuration Management</b> .....	<b>30</b>
Naming CIs .....	30
Availability Status for CIs and Services .....	30
Vital Business Services .....	31
Baseline .....	31
DSL and DML .....	31
Complex/Component CIs .....	31
Viewing affected Services from a CI window .....	31
Development Environments .....	32
Reporting on Configuration Items .....	32
<b>Service Level Management</b> .....	<b>33</b>
<b>Service Portfolio Management</b> .....	<b>35</b>
Service lifecycle .....	35
Continual Service Improvement .....	36
Self Service and Service Catalog .....	36
Managing Multiple Services .....	37
Defining Service Levels .....	37
<b>Working with External Data Sources for Advanced ITIL Activities</b> .....	<b>38</b>
Connection to external monitoring tools .....	38
LANDESK Management Information .....	38
<b>The Design Ideas Database</b> .....	<b>39</b>
Incident Management .....	39
Problem Management .....	42
Change Management .....	47
Request Fulfilment .....	52
Event Management .....	57
Service Portfolio Management .....	59
Configuration Item and Asset Management .....	62
Release and Deployment Management .....	63

# Design Ideas

The LANDESK Service Desk design components enable you to control the on-going design of your Service Desk system and so implement a wide range of approaches to match your service management requirements. Often there is more than one way to achieve the same result, and the best choice often depends on your own business and how you want Service Desk to help you to run your IT and service departments.

The IT Infrastructure Library (ITIL) provides a rich set of guidelines and best practices around the IT Service Management operation. Service Desk is designed to help you to follow the ITIL guidelines, but also enables you to modify and implement your own version of ITIL processes and activities.

This document gives recommendations and suggestions for design changes that you can make to the standard implementation of LANDESK Service Desk using the design and configuration tools that are provided as a part of the application. Much of the content of this document is provided by customers, implementation consultants and product experts.

**NOTE:** This document refers to some terms and concepts as described in the IT Infrastructure Library V3 (ITIL 3).

**NOTE:** This document is for people who are familiar with LANDESK Service Desk design and administration tools. For more information about these, see the *LANDESK Service Desk Designer Guide* and the *LANDESK Service Desk Administrator Guide*.

**NOTE:** Further suggestions are also available from the LANDESK User Community (<http://community.landesk.com>).

You can find advice and suggestions about:

- *General Design Considerations*
- *Availability Management*
- *Capacity Management*
- *Change Management*
- *Event Management*
- *Financial Management*
- *Incident Management*
- *IT Service Continuity and Disaster Recovery*
- *Knowledge Management*
- *Problem Management*
- *Release Management*
- *Request Fulfilment and Service Requests*
- *Service Asset and Configuration Management*
- *Service Level Management*
- *Service Portfolio Management*
- *Working with External Data Sources for Advanced ITIL Activities*
- *The Design Ideas Database*

## General Design Considerations

1. Ensure that every action is audited with a timestamp and a record of the User who committed it. See the *LANDESK Service Desk Designer Guide* for how to add these attributes to your records. Best practice is to include these fields on all of your windows so they are easy to reference.
2. Furthermore, if a field should not be altered once it has been saved, use window design rules to ensure that it has the write once property.
3. Ensure that fields you need to have completed are set to mandatory on your window designs.
4. Use group and role privilege structures to ensure that only authorized personnel can perform the various actions available on your processes. For even greater control, you can also make individual attributes privilegable, so even if they appear on a window that someone can see they may not also have privileges to see the attribute itself.
5. Ensure that all users defined in the system have (as a minimum) their contact details and location recorded against their record in addition to their name and customer group.

# Availability Management

There is no single prebuilt process provided with Service Desk that directly delivers Availability Management. However, Service Desk does support you in performing the various Availability Management activities driven through your ITSM processes.

For example, a key concept in Availability Management is the setting of target availability levels for IT Services, and the on-going review of the actual availability of live services against those targets. Using the Service Portfolio process (Service Portfolio Management on page 35), you are guided to identify the availability levels required for each Service during the Design stages. These are recorded on the Service CI or in individual Service Level agreements. Then, using Service Desk's ability to connect to external sources, you can view or import actual measured availability statistics for CIs or Services from external availability monitoring tools into the latest-value attributes on the Service CI.

## Cost fields

The definition of a Service-CI includes a **Cost** field, presented to the end-user to indicate the departmental charge for receiving that Service, and also an asset cost / purchase cost. Using Object Designer, you can add a further **Cost** field to describe the cost to the business of any downtime. You can calculate this value yourself based on the nature of the service, its usage and its purpose in your business. Also relevant to Financial Management, these definitions in Service Portfolio and Configuration Management are direct feeds into the calculation of Service Unavailability as seen on your Service Availability reports.

Another process that impacts the monitoring of Service Availability in Service Desk is Event Management. Critical Events are automatically created and resolved as devices delivering services become available and unavailable. This provides easy tracking of uptime, downtime, mean time between failure, mean time to repair and so on. Use the provided Crystal Reports to see these results.

**NOTE:** For more information about Event Management, see the *LANDESK Service Desk Event Manager Guide*, and *Event Management* on page 13.

## Reporting on Availability

All data imported into LANDESK Service Desk can be reported on for use by all the other processes followed in the tool.

- To get an overview of availability by CI Type, create a query over all CIs and group this by type. Their availability data can be used as columns on the results set – such as the Outage type, the device down time and device up time.
- To understand the impact of failures to Vital Business Functions (VBFs), write a query over all CIs and filter this by those that are marked as a VBF. Opening these records enables you to access the impact diagrams associated with these services.
- To report on the Serviceability of suppliers, use the **ServiceCapacityPerformance.rpt** Crystal Report.
- A single point of failure (SPOF) report is included out of the box for identifying items in the CMDB that may need attention.
- To conduct a component failure impact analysis (CFIA), use the impact analysis feature in Configuration Management.

## Capacity Management

As with Configuration Management, there is no single prebuilt process provided with Service Desk that directly delivers Capacity Management. However, Service Desk does support you in performing the various Capacity Management activities driven through your ITSM processes.

For example, key to Capacity Management is setting target capacity and utilization levels for IT Services, and the on-going review of the actual capacity and utilization of those live services against those targets. Using the Service Portfolio process (see *Service Portfolio Management* on page 35), you are guided to identify the capacity utilization values required for each Service during the Design stages – effectively sizing the service or application for the maximum number of anticipated users. These are recorded in the **Capacity Target** fields on the Service CI. These always need to match or exceed the Service Level capacity targets if any are identified in the Service Level agreements recorded on the Service.

## Tests and Risks

---

The portfolio process provides you with a collection to track Capacity Tests and their Results, or identified Risks and their Countermeasures.

## Planning

---

Producing a Capacity Plan is enabled through the Service Lifecycle, where the Capacity Manager is prompted to create and then link the requirements, scope and forecast for the service together in a Capacity Plan document. You can access this from the attachments on the Service Lifecycle. The activities in creating this plan are largely manual, but access in Service Desk to all the service target – and captured – values on the Service and Lifecycle process helps the Capacity Manager with this. Also refer to the information produced from Reports to input into future planning.

Service Desk can access the data in many external applications, silos and systems, including capacity management / monitoring Information systems. This means that, using Service Desk's ability to connect to external sources (using the Data Connector and Import components), you can view or import actual measured capacity values for CIs or Services from external capacity monitoring tools into the latest-value attributes on the Service CI. As with Availability Management, this is not limited to capacity values alone, but can include performance, usage and financial data.

**NOTE:** For more information about connecting data sources, see the *LANDESK Service Desk Administrator Guide*.

Reporting on Capacity is a key part of this activity. The collection of Capacity Data on the Service CI gives access to capacity values from queries. You can, for example, add Trend charts of CPU utilization over time to Dashboards to show trends in these captured values.

Using Crystal Reports (or other reporting tools) can help greatly with the production of management reports from historical capacity data in Service Desk. Sample Crystal Reports are provided to show the tracking of capacity performance against targets over time. Exceptions and forecast values are presented as calculations based on the Target and Actual values stored in Service Desk. These can be seen tracking sample metrics for Service Performance numerically and graphically. You can also use these views of captured performance to help predict and forecast future directions of Capacity attribute values.

The captured capacity data is presented at all times as a list, either as the **Capacity Data** tab on the Service or as separate Service Capacity queries. This provides access to an even more advanced analysis, since you can export the query results to Excel, from where more detailed analysis tools are available.

**Design Idea:** Consider using LANDESK MI to re-capture and present these values as Trend gadgets against defined thresholds.

Another key capacity metric is the number of users subscribed or linked to the Service, which again can be monitored over time.

**Design Idea:** Although you can manually update subscribed users, following Change or Request processes could also update the Service automatically.

Customer Utilization data is a capacity metric that is also typically provided from external sources. A daily import of maximum concurrent users, or current user count for an application provides comparison and trend abilities against targets for application sizing. You can show trends in these captured values by adding charts of, for example, human usage over time on Dashboards to help with forecasting capacity requirements.

## Simulation, Modelling and Forecasting

---

Use any of the above Trend examples to monitor the growth of utilization and reduction in capacity. Any of these Trend charts also allows extrapolation to forecast where and when limits may be breached in time, and to consider potential future points where demand may exceed the maximum target set.

You can address sizing and modelling in Service Desk by working with the CI Structure diagrams that define the service relationships. Create a new **Model Services** folder in the CI Structures section and build your example model services there. You can reuse the key concept of a **Services** relationship to build 'model' examples. You may also choose to create an alternative relationship (Service Model). Your Capacity reports should then pick the Model structures and provide modified roll-up capacity information and forecasts based on these example structures.

You can simulate adding, for example, a different server to a Model service and compare in the Capacity Summary Report the differences between the Live and the Model services. These changes can be seen in forecast reports and enable you to simulate changes and variations in existing services before the changes take place.

# Change Management

The purpose of Change Management is to ensure that all Changes are made in such a way as to minimize their effect and the effect of any related Incidents on the business.

## Emergency, Normal and Standard Changes

ITIL guidelines describe three different Change processes that you could use. With LANDESK Service Desk, you can create as many different types and variants of Change processes as you require – there is no limit to the number of Change processes that you can design using the design tools. The default Change process is comprehensive as a full Normal change process.

**Emergency Change** – You can modify a Change process to bypass most of the standard actions in the case of an Emergency Change being logged. You could do this by adding a new boolean **Emergency Change** attribute to the **Change** object, and then adding a process decision to route to the Change process past as much of the process as required.

**Standard Change** – The same approach applies to the simpler 'no authorization required' Standard Change activity. Adding a boolean and process decision again for **Standard Change** enables a third branch.

However, you could choose to use three separate processes – one for each type of change. The advantages of this are that you can maintain and develop the separate processes independently.

**NOTE:** During implementation, your Service Desk consultant can provide guidelines based around these approaches.

In typical, mature Service Desk implementations, Change processes grow and evolve way beyond the simple ITIL definitions. It is common to have a large number of variants of change processes.

**NOTE:** In considering your different change processes, note that the **Change Type** categorization list is predefined for you with **Major**, **Minor** and **Significant** types.

## Change Schedule, Forward Schedule of Change, and Projected Service Outages (PSOs)

Communication is a strength of LANDESK Service Desk. Many organizations communicate future and planned activities in different ways, but whether referring to them by the ITIL terminology or other terminology, the same end result is easily achieved.

**Viewing a Change Schedule** – At the simplest level, the correct use of Queries can provide everyone with the correct views of scheduled work. You can create a Change Schedule query based on the Change Schedule object. We recommend that you use two versions of this query – one that can Launch to allow Change Managers and other authorized roles to access the full RFC/Change, and the other to provide other departments such as Incident Management / firstline staff and end users with a read only view of any planned Changes.

**NOTE:** Remember that you can schedule these Future Change queries, e-mail their results, and also provide them using RSS feeds. For more information about using RSS feeds with Service Desk, see the *LANDESK Service Desk RSS Generator Guide*.

Remember that you can add the queries described above to Self Service for your customers or end-users to see, and can be made rich in appearance and content using Report Templates in Object Designer.

**Projected Service Outage (PSO)** – If you need a PSO view presented within IT or to the business as a whole, then only one small modification is required to the standard system design. Include the **Title** of the impacted Service on the Change record. Depending on your initial design, this may need only a dropdown list of CIs filtered to show just those where the CI Type is Service.

You can use a new Object inside the Change process to record multiple **Affected Services** on a Change, however an easier approach may be just to use a text field on the Change window itself. Different Service Desk implementations tend to approach this in different ways based on their own Service-lifecycle maturity.

Once added to the Change, a new variant of the Change Schedule query can be presented through all platforms to list future scheduled Changes that will impact the availability of Services.

**NOTE:** We recommend adding Projected Service Outage queries to Self Service, using HTML Report Templates in Object Designer to add visual impact. See the LANDESK User Community (<http://community.landesk.com>) for examples of Self Service design.

## Software Development and Enhancement

LANDESK Service Desk is well suited to designing processes beyond generic IT changes. Many organizations use Change processes across a range of product and service lifecycle activities, including Software or Application development and maintenance. A good starting point for this is the Release Management process provided in the standard Service Desk database.

## Change Type and Categorization

If you have one Change process with branches for different types of Change, you add a dropdown list to the Change window from which you can select the **Type of Change: Emergency, Normal, or Standard**. Selecting **Emergency** routes the process through **Emergency CAB approval**; selecting **Normal** follows the full Change process; selecting **Standard** follows a flow that requires no authorization or pre-approval activities. These options are either on the Change window itself or provided as three separate shortcuts in the shortcut bar.

To help you to identify your changes, you may have a dropdown list of the types of **Change Categorization** that are available. These are **Minor, Major, or Significant**. Select these from the dropdown list on the Change Window to further identify your changes.

## Approval and CAB

The **Approval** stage on Change processes varies both by the type of Change being followed, and your own organization. Make sure you publish the appropriate shortcuts to the each of the roles and groups in your Service Desk. With Service Desk, a default role of **Approver** is provided, and you can create a Support Group called **CAB**. You can add your own individual roles within the **CAB** and your own alternative CAB groups through the **User Management** tree in the **Administration** component.

When a Change reaches an approval stage, add a notification or dashboard to update the members of the CAB group. Using the Process Approver functionality, they can be sent an e-mail containing links to the Change details, from where they can click **Accept** or **Reject** in the **Actions** list.

**NOTE:** Alternatively, you can design a dashboard or query for the CAB group that shows the Changes they need to progress.

**NOTE:** If you want, you can add individual weighting calculations into the process.

**Design idea:** If you are holding a face-to-face CAB meeting, run Service Desk in the meeting (with large screen/projector) and accept/reject approvals during the meeting.

## Backout plan

---

There are many documents associated with good Change Management, and Service Desk's strength lies in its ability to guide people to produce and attach those documents to the Change process. To add an attachment to a Change, click **Add Attachment** in the **Actions** list.

**NOTE:** You can add **Add Attachment** as an optional action to any status using **Process Designer**.

However, in the case of specific structured activities such as Backout plans, you may want to use Service Desk to ensure that the documents are also correctly progressed and acceptable.

For example, you could add an **Add Backout Plan** action to the Change. The Add Backout Plan window would contain the Attachment option for the document, but also fields to validate the content. You could also add a dropdown list of Backout Plan Statuses, using a reference list. For example, you could add values of **Submitted**, **Approved**, and **Rejected**.

**Design idea:** The process-driven nature of Service Desk means that you can extend this further. You can create complete assign, create, review, approve, reject process sequences either as branches on your Change process, or as Change Task processes.

## Change Review

---

The default Change process contains a **Change Review** stage and status. Once the Change is Implemented, a Change Review takes place – the process is assigned to the appropriate role with a notification that a review is required. Although comments and notes can be added at this status, you could expand this into greater review activity.

**Design idea:** Add further process activities and actions (assigning to more users, roles and groups, or adding extra review/reject steps). Or use an automatic action to add a customized Change Task process at this stage, and add a Precondition to the Change that prevents it from being closed until the Task is Completed.

# Event Management

This section gives ideas for how to use the Event Manager component with LANDESK Service Desk.

## Using Event Manager for Comprehensive Event Management

The Event Manager component in Service Desk enables you to configure external systems to create, progress and resolve/close Event processes in Service Desk. This section highlights best practice and gives design ideas for some key concepts.

**NOTE:** See the *LANDESK Service Desk Event Manager Guide* for more information about the configuration and usage of Event Manager.

Any tool that can generate a URL, a command line, or a batch file (with parameters that identify the type of event and the CI affected) can interact with Event Manager. Typically, these are monitoring tools that can send alerts (for example, by e-mail, text message, URL, or custom executable) when an alert is required.

However, do not configure Service Desk to respond to every alert that a monitoring system generates: use Event Manager for significant alerts only. Event Manager is not itself a monitoring tool, but is a consolidation point where significant alerts can be converted using Event processes.

Event Manager can receive alerts from many different sources, and can map different sources to different Event processes – remember that you can use many different processes with Event Manager. For example, you could configure a Major Event process and a Standard Event process. It is not practical to monitor everything to a detailed level: we recommend that you create an Event process only for those events that impact the delivery of a service. You can include information and warning alerts, but consider if this provides too much noise before you implement this.

The URL or batch file / command line from your monitoring tool can contain many attributes that populate the initial Event process. These typically include the Event Source, Event Type, CI, and Title. However, you can map many other attributes, such as a Priority and/or Response Level to provide you with escalation activities (color changes, e-mail notification, reassignment, and so on).

## Linking CIs to Services

If you link your monitored CIs to the Services in the Service Desk CMDB, you can create a query on the CI relationship table, linked to the Event processes to see where CIs are recorded on active Events, and also see the Service that is affected/connected to the CI on the Event. With these relationships in place, you can run an Impact Analysis from an Event processes.

## Auditing Events

All process actions are available to Event Manager – not only Create / Add Note / Resolve and Close. You can create and use custom actions to use with Event Manager that are based on the type of alert or notification that your external system can generate. These actions are recorded and audited like any other actions, so by associating an Event Manager system user with the Event Manager service, you can identify which process actions resulted from external systems monitored by Event Manager.

## Notifications

Notification is the same as with any process. In your default Event process, always include an assignment with relevant notifications. Also include escalations using Response Levels to provide many other notification options.

## Create a Current Events query

---

Add a Current Events query to Incident Management dashboards to enable Incident Management staff to see, in real time, where current Service delivery may be impacted. If you are using Self Service, you could present this information to your end-users graphically (perhaps like traffic lights) for the key business Services. When an Event process is created, set the status on the CI or Service to Impacted, and present the relevant image from the Status on the dashboard.

## Automatic and manual actions

---

When developing your Event process designs, remember that an Event may involve both automatic and manual interactions. You may not want everything to be automated, but instead to use analysts to validate certain steps.

For example, you can configure Event Manager to create an Event process that automatically appears on an Event Management dashboard, and is automatically escalated to notify an analyst. The analyst then reviews the Event and decides whether to mark the affected Services as impacted, and whether to generate a linked Incident from the Event. All of the standard process-linking abilities remain, so you can create a Major Incident, generate a Change, and link an Event to a Change within your process design.

## Automated recovery

---

You could create an Event process in Service Desk that includes all of the required ITSM notifications and escalations. However, you can configure the LANDESK Automation to perform a recovery action when it detects the event (everything from a rebooting to starting a new virtual environment). When the monitoring tool detects that the device/service has recovered, it updates Event Manager, which then closes the Event process. All of the availability statistics for the Service are captured, but the downtime is minimized.

## Reporting on Event Management

---

When you are automatically creating and resolving Event processes, you can produce some powerful reporting and information outputs.

While you are creating Events for system-down type occurrences, you are automatically recording the downtime and uptime of a CI and potentially also the service it is attached to. You can amend the Mean Time Between Failure reports supplied with Service Desk to report on this information.

A graph of Events over Time for one or more Services is a powerful input into Problem Management, providing awareness of any increased rate of events outside of the system monitoring team.

## Event Manager mappings

---

Event Manager provides mapping of fields from incoming events to new processes. For details of how to set this up, see the *LANDESK Service Desk Event Manager Guide*.

Typically you will map the **Event Description**, **Configuration Item**, **Priority** and the date time that the Event process was created.

To map these fields follow the instructions described in the *LANDESK Service Desk Event Manager Guide*, but make sure that you link the inbound **Event Description** to the **Description** attribute on the Event process. Do the same for the **Configuration Item** and the **Priority** if one is provided. You do not need to create a mapping for the Creation date and time, because this is set automatically.

Using the Service Level Agreement Rules enables you to set the Response Level automatically from the **Priority** or any other values on the Event. This means that the escalations and notifications from alerts can vary based on the Event's priority.

If you want lower priority Events to be routed and progressed differently from high priority or critical Events, modify the process to include decisions and branching based on the **Priority**. Create a decision on the process, and drag the actions you require onto each path out of the decision. Alternatively, create three different Network Event Sources for different levels of alert, and configure each to launch a totally different process.

**Design idea:** it is easier to manage a single Event process with branches than to maintain several similar processes.

Any process action can be performed through Event Manager alerts. However, there may be some that you will always need in addition to creating new Event processes: for example, Resolution or Closure.

### To Close or Resolve event processes from alerts automatically:

1. Set up your Event as usual.

**NOTE:** For more information, see the *LANDESK Service Desk Event Manager Guide*.

2. Select the action **Close** or **Resolve** as appropriate.
3. Complete the mappings.  
Usually you map only **Event Description** to the **Description** on the Closure or Resolution.
4. Create a suitable event definition.  
Usually you specify that an Event with Type of Close performs the Close action set above.

You may want some actions to run automatically, based on the Type or Priority of the Event. For example, if a Priority 1 event always needs a linked incident to be created (for example with a Major Incident), then you can use the process branching described above. On the high priority branch of your process design, add a **Create Incident** automatic action with related attribute values, so the name of the **CI**, **Service** or **Description** is automatically copied onto the linked Incident. You can also set default values on this automatic action to preset values on the new Incident.

The same creation behavior applies for Changes. Although you can manually create a Change from an Event, you could automatically create, populate and link a Change in Process Designer in the same way as described above.

## Handling different Event types

Monitoring tools provide many different types of alert. Within Service Desk you can split different alert types into different Event processes to ensure the appropriate behavior. Use the **EventType** passed in from the alert to route the Event process through different branches – to create a major Incident from a failure, for example, or to provide notification-only warnings and no action for information. Create a Decision in Process Designer for **Is Event Type = Information** and **Is event Type = Warning** to route down the correct Event path.

Alternatively, you can have different alert types initiating totally different processes. Typical alert types include Information, Warning and Exception, which you can set manually on the Event window.

### To trap and process Information, Warning and Exception alerts as separate processes:

1. Create a default Event Type called **DeviceDown**.
2. Follow the guidelines in the *LANDESK Service Desk Event Manager Guide* to create new event sources for the Event Types **DeviceWarning** and **DeviceInformation**, (and possibly **DeviceException** if this is different from **DeviceDown** in your organization).
3. Map each EventType to the relevant Event Process to provide the behavior required from different types of event.

**Design Idea:** Don't bring too many alerts into Event Management. Your monitoring tools serve a purpose, and bringing every information message and warning into Service Desk processes may be unnecessary. You can use browser controls on dashboards and windows (in Console) to display web content, so you can often display your monitoring tool's full set of alert and notification types directly in this way, inside the Service Desk interface.

# Financial Management

This section gives ideas for how Financial Management activities can be applied with Service Desk.

## Financial Management Overview

No single process or product feature delivers this ability as a 'package'. Instead, a combination of activities working with the Service Lifecycle process (see *Service Portfolio Management* on page 35) enable you to achieve Financial Management of your IT Services.

To achieve these Financial Management suggestions, you must be using at least the following activities and processes in Service Desk:

1. Configuration Management
2. Service Portfolio Management
3. Service Catalog Management

When working with financial data in Service Desk, you may choose to hide windows and fields from general visibility. Financial data can be confidential and it may be inappropriate to share it with IT staff. You can make fields or windows in Service Desk selectively visible by setting privileges on individual fields in Object Designer, at a Business Object level, or by publishing different window views to different groups or roles.

**NOTE:** For more information, see the *LANDESK Service Desk Designer Guide*.

## Cost Allocation in Service Desk

In Service Desk, individual Services are made of linked component CIs. Delivery of each Service has associated costs that can be simple or complex depending on your needs.

To record costs for the Services, add the Service Cost fields in Window Designer. These include a total cost value for the Service and a cost center list.

As Services and component CIs are all CIs in the CMDB, these fields are available to apply to any CI as well as the top-level service. The Cost values vary according the type of CI: the costs for hardware items include the hardware costs and the maintenance costs; software costs may include both the licenses and maintenance. Additional costs such as personnel and utility may also apply.

Consider also the different cost categories that may apply. When adding a cost, set the appropriate category type value on the cost to indicate if it is direct, indirect, fixed, and so on.

In a shared service model, you require more than one cost center or business unit to have costs allocated to it, you can also change your cost design in Service Desk. Add a Collection of Costs to the Service, with each Cost having a value, a cost center or business unit and the different cost types as described above.

Other Financial Management concepts in Service Desk are:

**Rolling Plans:** During the evolution of a Service through its strategy, design, transition, and operation stages you may want to view the associated costs. Modify your Service Portfolio query to show the current cost for each Service alongside its portfolio status. When your service costs are understood, it is also useful to see a view of future expense over time for a service to help with producing an ongoing Rolling Plan. Refer to the example financial reports provided to see both the impact of downtime Costs over Time, and also the Service income and expense over a three year period.

**Metered and Variable Costs:** Where a service is supplied on a per-user, or per-time-period basis, your Finance reports can multiply the defined service charge (on the Service CI window) by the appropriate variables such as number of users or periods of time.

## Customer Charging

---

In addition to the various costs that IT may bear in delivering Services to the business, you can also apply charging to the customers of IT for the services that they consume via request fulfillment. Different charging models can be supported in Service Desk:

**Service Level Management:** If you are using Financial Management activities it is very likely that Service Level Agreements are in place between you and your customers. You can detail different tiered customer charge models in the Gold, Silver, and Bronze descriptions on the Service Window. You can define additional charge options beyond these examples, but good Financial Management always benefits from following the simplest model possible.

**Notional Charging:** Where no actual transfer of funds, payment of bills, or balancing of funds takes place, you can still charge 'notionally'. That is, the full extent of the charging model can still be followed.

**Direct/Cost-Plus Charging:** Where a combination of Direct and Indirect Costs may be applied, the financial reports produced can add these together providing a 'direct-plus' view.

**Fixed Charges:** This is a simple calculation where for each customer a price is defined for the service. Again, the Financial report used provides the calculation for this.

**ROI, NPV and IRR:** The ROI and the values for Net Present Value and Internal Rate of Return are calculated with Service Finance Reports. Enter the current total cost of service produced from your Finance Cost reports in the Cost of Service field. Enter the current total service earnings from your Finance Billing reports in the Service Earnings fields. Calculations from within Crystal Reports produce ROI, NPV and IRR. For more detail on the formula for these, see the Crystal Reports help files.

Post Program ROI is not calculated as a function of Service Desk. However, Service Desk does support the softer activities that typically make up the function, such as questionnaires and surveys. To ensure that this takes place, add an action to the Service Lifecycle process to gather the necessary data. Place this action after the Go Live of the Service. Adding this for all your Services should ensure that every Service can be assessed this way, in a consistent fashion.

## Budgeting

---

In order to understand if your services are being delivered in budget you will need to record both how much money you are expecting it to cost and how much it actually does cost in terms of delivering the service. To see how far apart these numbers are, use the BudgetDeviation.rpt Crystal report provided.

## External Data

---

As with most other activities in Service Desk, you can import data from multiple sources to populate cost fields on the Service if required. For more information about data import, see the *LANDESK Service Desk Administrator Guide*.

# Incident Management

The purpose of Incident Management is to restore normal service operation as quickly as possible with the minimum disruption to the business.

Service Desk contains processes for all your ITSM support processes, so you can give Support Analysts visibility into other processes to help them in their work. The most typical example of this is a view of Problems presented to the Analyst working on an Incident. You can do this in a number of ways:

- Make Problems and Known Errors 'knowledgeable' so that the Incident Analyst can search for Problems and Known Errors that match the Incident from the Search toolbar.
- Add a tab to the Incident window that displays Problems matching the category (or CI). When the Analyst selects the required CI or Category on the Incident (or any other Process window), the tab populates showing Problems matching the CI or the Category. You set this up using a Filter query, which has the advantage that you can restrict this to only active Problems if required.

## Creating other processes

Most Service Desk processes include actions that enable you to create other service management processes. For example, on an Incident at the **Open** status you will often find **Create Change, Create Problem, Create Request** and so on. This is not limited to these titles – you can launch any support process from these actions. Click the required **Create...** action and the required window for that process appears. When you save the new process, it is attached as a Child in a folder on the process tree, called **Child Incidents, Child Problems, Child Changes, Child Requests**, and so on. When they are linked, the child processes also contain links back to the parents.

**Design Idea:** to optimize this process, you can create a new action as a part of the Incident process (for example, **New Request**) that presents only a 'summary' set of fields to the analyst. However, when saved, an automatic action populates the target process with the fields created from the summary action. This helps to stop Analysts in one area from having to work with and understand complex windows designed for other parts of the organization.

# IT Service Continuity and Disaster Recovery

This section gives ideas for how Disaster Recovery and Continuity planning activities can be applied with Service Desk.

## Continuity overview

---

No single process or product feature delivers this ability as a 'package'. Instead, a combination of activities working with the Service Lifecycle process described in detail in Service Portfolio Management make it possible to plan and follow Service Continuity practices in LANDESK Service Desk.

To understand the Risks of continuity issues, use Impact Analysis feature of Configuration Item management to view the impact of any CI in the infrastructure going down, including the ability to see exactly which users will be affected.

## Test Planning

---

The Service Portfolio and Service lifecycle is key to Service Continuity Management activities in Service Desk. Following the Service Lifecycle process enables you to build and complete continuity plans for each IT Service as a formal part of the Service lifecycle. These plans can be distributed to relevant personnel by means of granting them access (via privileges) to the parts of the Portfolio record that pertain to continuity.

Once a Service is live, a collection of Test or Scenario records is available.

Since each Service comprises a number of component CIs, you can run an impact analysis at any time by right-clicking the Service then clicking **Impact Analysis**.

In addition, each Service in Service Desk contains a number of continuity-related fields. If these are not on your Service window, create them using Object Designer and Window Manager. Fields added for Continuity Management include:

**Minimum Recovery Time.** Although a field on the Service window, consider also as a Response Level applied to a Major Incident for that Service.

**Business Criticality** on the Service Window enables you to specify a relative criticality of the Service. Add this field to the Service Portfolio query to list your Services and to group and order your Services by criticality.

**Recovery Option** on the Service CI window describes the options for recovering this Service. Add this field to the Service Portfolio query to see recovery options for all Services and vital business functions.

**Cost of Downtime:** used also in Availability Management, the **Cost of Downtime** field on the Service enables a financial value to be recorded against a Service.

The Service lifecycle in Service Desk enables Continuity Management staff to add a collection of continuity activities – where a Test contains a test description and its result, and a Scenario allows for continuity scenarios to be recorded with a proposed resolution.

The **Continuity Test Due** field on the Service window enables you to set future test dates and select when they are due. Add this field to the Service Portfolio query to list your Services and to group and order your Services by **Continuity Test Due**.

Detailed continuity plans are attached as attachments to the Service-CI and can be accessed as required by appropriate staff from the Service window.

**Design Idea:** If you publish Major Incidents through Self Service and Web Desk, provide a window behind each record so that you can access the attachments from there. This enables those with the appropriate permissions to access the continuity plans directly from the published alert.

Although plan documents are not typically version-controlled themselves, you can make a document version controlled in Service Desk by using a **CI Type of Document**, and attaching the document to the CI. Then manual or automatic CI-versioning keeps the correct version stamp and history of changes, and RFCs can be followed to implement document version changes.

**Continuity SLAs.** The Incident process in Service Desk provides a Major Incident status. You can set alternative levels of response and service that apply when a service is in a major incident / disaster status by defining a new Response Level that applies automatically for that Status.

Some contracts and agreements are critical to ensuring delivery of the service. Set the **Critical Contract** flag on the Agreement for the Service to identify these contracts.

To view reports and information about continuity plans, test results, and other related information, modify your Service Portfolio queries and dashboards to show the continuity information captured for each Service.

# Knowledge Management

The Article process in Service Desk provides mostly the same abilities as seen across all other modules. This section describes some knowledge-specific ideas and suggestions that may help in designing and using knowledge.

## Numbering Articles

---

If you want unique numbers on articles, add a **Unique ID** or **Reference Number** to the **Article** object as described in the *LANDESK Service Desk Designer Guide*. This number appears in the title bar of the Article and can also be seen in queries when included.

## Knowledge relevance

---

Knowledge relevance is a value shown in search results to confirm how well an item of knowledge matches your search terms. In your search results, you can also show an alternative form of relevance: to whom and when is this knowledge most significant.

**NOTE:** For example, an item of knowledge about correcting the output of financial reports may be particularly relevant at the end of each quarter when those reports need to be run.

The simplest approach is to add a text field on your Article to hold this unstructured text, and then present it as a part of the search results.

**Design Idea:** You could add a **Project** field to Articles. When you create an Article, use this list to select the project, activity or reason why the knowledge is created. If the **Project** itself is driven by a Change process, you could automatically pre-populate this link as described elsewhere in this document.

## Service-centric Knowledge

---

You can add a dropdown list of CIs to the Article window. Then, when you create a new article, you can select which CI the Article applies to.

Alternatively, you could use **Keywords** or **Tags**. Also consider the Knowledge-centered Support methodology and the use of Knowledgeable Words.

If you are using a full Service Lifecycle, your Services are defined as CIs and appear in the same list. You can change this list to show only Services and then flag each article with a Service.

## Validation and verification

---

Every business has different needs for validating the content of knowledge. Service Desk Knowledge is process-driven, so you can include the same form of validation and approval stages in the pre-publication of knowledge as for any other process.

To help with this process, fields that you add to the Article window can ensure the correct information is captured. Complete validation fields such as SOX / HIPAA / ISO and other governance requirements. Remember you can report on knowledge just like any other process, so a report of Articles showing Governance fields and Review Dates can help with your external audits.

## External Links

---

Frequently, knowledge that needs to be shared with users is explained already on a webpage or other external source. Links to these sites can be stored on your articles and can be launched from within the articles by using desktop management capabilities to launch the URL in the field.

## Expiring Articles

---

Articles only appear in knowledge results when they are at the **Approved** status. However, sometimes Articles need to be expired when they are no longer relevant. This is achieved by setting the **Expiry Date** on the Articles. When the Knowledge Management Engine service runs, it removes expired articles from the index so they no longer appear in the search results.

## Known Errors

---

In LANDESK Service Desk, the knowledge management features provided lend themselves particularly to the management of Known Errors generated from Problem Management processes. In this section we describe how these items fit together.

**NOTE:** For more information about generating Known Errors from Problems, see *Problem Management* on page 25.

Known Errors are generated during problem diagnosis. One of the benefits of Service Desk is that by making Known Errors 'knowledgeable' (that is, appearing in searches), you can ensure that they appear in searches from all other processes. Knowledge is shareable across all modules and usage in Service Desk. This means you can search for Known Errors from any process or window – even from a CI or Service window if required.

### To find a known error matching an Incident, Change, Request or any other process:

1. Click the Search button on the toolbar.

**NOTE:** If dynamic searching is enabled, a shortcut appears in Console.

2. When the matching Known Errors appear, right-click the required search result, then click **Attach**.

## Workarounds

---

You may see workarounds described in two places in Service Desk. Most typically, a Workaround is created as part of a Known Error in the Problem process.

### To add a Workaround:

1. In the Problem process, click **Create Known Error**.  
The Known Error window appears. It contains both an **Error description** and a **Workaround** field.

**NOTE:** You can make the **Workaround** field mandatory if required.

You can also capture Workarounds at later stages in the Problem process. You could add a **Workaround** attribute as a text field on the **Problem Resolution** object and place it on your Problem Resolution window. Then, when resolving the Problem, the Workaround can be recorded.

Remember that this is the workaround identified during the problem, and recorded when the problem is resolved, so it may be valuable if that problem itself is ever encountered again. This is distinct from a Workaround on a Known Error (as described above), which provides advice on managing the error/symptoms of one or many Errors that may occur from that Problem.

## Using Knowledge throughout the Service Lifecycle

---

Because of the way that LANDESK Service Desk is designed, a knowledgeable item is available to search for regardless of the status of the service to which it refers. Therefore, when knowledge is created about a service during its design phases it will continue to be available to search for when the service transitions in to chartered and even retired statuses. If it is not always desirable to have knowledge available to all users then domains can be used to limit this.

The knowledge index can also be used to present results of all the other processes in the IT environment. Any attribute on any object in LANDESK Service Desk can be made knowledgeable as can any document on the network (including doc, docx, xls, xlsx, pdf, ppt, pptx etc) and all these different items can therefore appear in search results. As a consequence of this, all data stored around any of the other IT processes can be made knowledgeable, including Portfolio, Availability, Configuration Items & Assets and any other processes.

## Further uses for the **Is Knowledgeable?** property

---

You can set the **Is Knowledgeable?** property to **True** for any business object using the Object Designer component. It is often a good strategy to start with only Articles available in searches, because until you are logging and resolving Incidents that provide useful and searchable information, you may find that indexing all of your Incidents provides more noise than value. For this reason, you could design your Incident process to create an Article automatically from an Incident Resolution when a **Create knowledge** check box is selected on the Resolution window.

**NOTE:** For more information about this, see the **Full Incident** process in the *LANDESK Service Desk Prebuilt Content Guide*.

However, the ability to include any data field from any process in knowledge provides more design opportunities than just searching Incidents and Known Errors. For example, if you are using the Event Manager interface for Event Management and Availability Management tracking, then you can set the **Is Knowledgeable?** property to **True** for your Event process descriptions. This means that you can search for event and availability information, and so create Articles from Event Closures. Similarly, you could add text fields to the Analyst window that describe the analyst's areas of expertise so that the names of Subject Matter Experts (SMEs) also appear in knowledge searches.

## Reporting on Knowledge

---

Like all records stored in Service Desk, Articles can be reported on. This is done by creating queries over the object or using Crystal Reports to generate more complex analyses of the data. Some useful reports to create and run include those listed below.

- **Articles by effectiveness** – List articles by their effectiveness rating. The top articles will help to identify trends in what users are struggling with and how articles should be written to be helpful. Review those with the lowest effectiveness to see if they should be retired or re-written to make them more relevant and useful.
- **Knowledge by service** – Write a query over Articles and group it by the service field.

# Problem Management

The purpose of Problem Management is to minimize the adverse effect on the business of Incidents and Problems caused by errors in the infrastructure, and to proactively prevent these things from happening.

## Impact, Urgency, Severity, Priority and Category

The typical ITIL assessment fields of Impact, Urgency, Severity and Priority on Problem records are already created and available. However, in the case of Problem, you need to create the Priority field as a new reference list in Object Designer. If you want to implement Impact, Urgency and Priority across multiple process domains (Incident, Problem, Change, Request, and so on) ITIL guidelines recommend that you use a common set of values (so the same values for impact, urgency, category and priority are used in all cases).

## Known Errors

Service Desk can provide Known Error recording and communication in a number of ways. The default design includes a Problem status of **Known Error**, which is usually sufficient for most ITIL implementations of Service Desk.

However, more mature ITIL implementations may choose to enhance Service Desk's handling of Known Errors. The procedure below describes how to use the standard design tools to implement a Problem process that automatically creates a knowledge article whenever a Known Error is created.

**NOTE:** For more information, see also *Knowledge Management* on page 22.

### To add Create Known Error to the Problem process:

1. Using the **Object Designer** component, create a new Object in the **Problem Management** module called **Known Error**. Include text attributes for **Symptom** and **Workaround** as well as the usual Creation Date/Time/User attributes. You could also add **Configuration Item** or **Service**.
2. Link the **Known Error** to the **Problem** object as a collection.
3. Optionally, add **Test / Development / Live** to the **Article Type** list (or as boolean checkboxes to the Problem window) to describe which environment the Known Errors were identified in.
4. At the appropriate (single or multiple) points in the Problem process, include the new action **Create Known Error** using the **Process Designer** component.
5. After the **Create Known Error** action, include an automatic action **Create Knowledge** that creates a knowledge article using the content described in the **Known Error** above.

## Other Problem processes

Although one process is provided to run through the diagnosis and resolution of Problems, other problem processes can be used in more creative ways to help with proactively identifying possible areas of future problems. For example, Fault Tree Analysis, an iterative process of considering possible causes of failure and possible components is well-suited to a different type of Problem process.

## Major Problems

A major problem is one that had a sufficiently high impact that it is worth reviewing afterwards if there are any lessons that can be learnt from it. You can include a step in the process to check if one is required by including a check box on either the Problem itself or resolution window and using this to route the process to an **Awaiting Major Review** status after the resolution.

## Release Management

The Release process in Service Desk behaves just like any other process – the same actions, queries and lookup behavior are available to you. For example, use the supplied process queries to see Releases by status, ordered by date, by unique identifier, or grouped by status, CI or Service as appropriate. In the same way, response levels, escalations, and notifications operate as with all other processes. When you create a new Release, as with any process, the Release is given a unique identifier, and the creation date/time and creation user details are recorded automatically.

**Design Idea:** Publish **Releases Due Soon** and **Recent Releases** into both your analysts' dashboards and into Self Service to communicate release progress automatically across the business.

In a Release it is typical to plan and schedule a number of activities, such as education, staff allocation, and communication. You can create these from the Release process using a Task. If you set a due date for this, consider using Reminders to warn that the work needs to be completed soon. Also add the **Due Date** fields to the **All Releases** query to provide dashboard/query visibility for these activities.

A number of items and activities around Change Management involve scheduling, so you can also pass your schedule queries into Excel for use with Project and analysis tools. Right-click any query in Console then click **Export**.

Some attributes on the Release process and window are specific to Release Management activities. You can add or remove these as required using Object and Window designer:

- **Risk**
- **Release Type**
- **Form of Release**

The behavior of your Release is controlled by these values – add Decisions as required to route down alternative process branches.

**NOTE:** For more information about Decisions, see the *Process Designer* chapter of the *LANDESK Service Desk Designer Guide*.

The Release process provides actions for the typical activities undertaken in a release; however, remember that every organization approaches releases in a different way. Some release processes are kept simple, whereas some manage every role and activity required. Where multiple activities take place in the life of the release, these are available to you from the Process Tree and the tabs on the Release window.

However, in the case of Release Management, linking Child Changes to an over-arching Release is typical. If you identify the Service as the CI on the Release, you may choose to add Changes with component CIs linked to the release. In this case, as the Changes are progressed and completed, the CIs are updated automatically using the Managed Versioning feature in Service Desk – the individual CIs as their changes complete, and the Service or released item when fully released to the business.

**Design Idea:** The Release process enables you to add linked Child Changes by default. You can also pause a Release, waiting for all Child Changes to be completed before automatically moving on. If you require your child changes to all be resolved or closed before the release can complete, add a Pre-condition to halt the process until **All Changes are Complete**.

**NOTE:** For more information about Pre-conditions, see the *Process Designer* chapter of the *LANDESK Service Desk Designer Guide*.

Preconditions must look for the status of child processes to be **Is Complete**. Place this Precondition before your roll-out activities in the Release. This enforces the requirement that all Child Changes are completed and confirmed before rolling out the release. Alternatively, add all of the relevant CIs for the Release to the Release process using the **Add Configuration Item** action. These can also be updated through the Release progressing to completion using Managed Versioning; however, a Precondition cannot apply in that case.

To include a further Release Review before deployment takes place, add a new process action **Release Review** or **Sign-Off** after the above Precondition. This adds the benefit of a review from an analyst after all of the Child Changes have been completed and before progressing to the roll-out stages.

When closing a Release, the categories presented for selection are, by default, the same as the standard Change categories. You can change these as required using the **Administration** component.

## Considering the bigger picture in Release Management

Mature Release Management often involves roles across the IT department and beyond – sometimes including external organizations. Some Releases require new CIs to be purchased, and some require existing CIs to be upgraded. Effective Release Management at this level is very hard to achieve in any system without following Configuration Management activities to build and maintain your CMS.

**NOTE:** See *Configuration Management* on page 28 for advice on how to build and maintain your CMS.

When planning a release it is important to understand the start-point of the relevant CIs to the release. This is achieved by importing data from discovery tools and using the discovered values for your CI properties as the baseline for the release. See *Baseline* for more information on how to create this.

Also, it is hard to progress without linking between Service Desk processes. As mentioned above, you can link Changes to Releases to coordinate and manage these activities. You can also create new Asset processes where required from Release processes. The Asset Procurement process and task are provided with Service Desk to assist with this task. Furthermore the release and deployment process is integrated with request fulfilment so that requests can be logged from the release process.

As with Linked Changes described above, you can use the status of these child activities to pause and move on the Release. Apply either a Precondition or a manual confirmation action in the process when waiting for procurement activities to complete.

The Release process acts as your control mechanism for ensuring that all of the correct activities in preparing the Release happened. You can also use the Release process to control the correct steps in preparing for deployment. Use Tasks in the process design to identify the required checklist of deployment technical steps, and then complete the Tasks to confirm that the steps have been followed. Design your Release process so that you can proceed with the Release only after all of the CIs relevant to the Release have been identified, located and accounted for.

As a part of preparing the Release, it is also important to consider the software license position. You can refer to the software defined in the DML (grouping in your CMDB – see *Configuration Management* on page 28) to confirm the existence of software, but you can use another Release task to actively confirm that sufficient licenses do exist to proceed with the Release from a legal position. If not, the Asset Procurement process can help with the steps to request, approve and purchase additional licenses to remain in a legally and financially correct position. Consider using a final legal/finance confirmation Task before the Release is approved.

## Request Fulfilment and Service Requests

The Generic Service Request process provided with Service Desk delivers a number of options for usage. Most of this process is driven from the values set on the Service that you select, so make sure that you correctly define your Services. One way to do this is to define a Service Lifecycle process for the life of your Services, and to ensure that the correct values are considered and reviewed before a Service is published. As a part of this process, add an appropriate **Service Description** that is appropriate to display both to analysts and to end-users. You could use TWO fields: one for an IT description and one for a Business description, and design your Windows and views so that analysts see both the IT and the Business description in the analyst interfaces, and the end-user sees only the Business description in Self Service. You can also include in this field instructions for how to access the service once it has been fulfilled.

**NOTE:** For more information about the Generic Service Request process, see the *LANDESK Service Desk Prebuilt Content Guide*.

When you are planning how to distribute and publish services, remember that the Service Catalog Administration component enables you to specify exactly which users, groups, and roles a Service is published to. This controls who can see the Service in the Catalog, but does not define whether or not the Service can be delivered to them: this is defined by the underlying process. For example, a member of the Design department may be able to request Technical Drawing software, but that person's manager may reject the request because the department does not have enough budget, or because that person has not yet been trained. Similarly, the license/stock team may pause or reject the request because there are insufficient licenses or because an additional purchase is required. This is all controlled through the Request process, so it is important to understand – and evolve – the behavior that your business requires.

You could choose to isolate Service Requests from the rest of the Service Desk, leaving Request Management operating separately from everything else. However, a bigger benefit comes from carefully ensuring that the Request is fully integrated with the entire ITSM activity view. For example, from Request processes you could:

- display a tab of all outstanding Incidents, Problems and Changes for the Requestor
- display a tab of all Events and availability information for the selected Service
- create new Incidents, Problems, Changes or other processes as actions in the Request process at the appropriate point.

A simple solution for this would be to add manual **Create Change** actions at various Request statuses. However, a better approach is to add an automatic **Create Change** action to your Request process at the place that a Change is required. For example, a Request for Access to a new virtual environment may require a confirmation that the current capacity of that platform can accommodate the additional usage. If more capacity is needed, then a Change is raised to request expansion of that platform. It is important that the Request is not fulfilled until that Change is complete. You can use Process Designer to design your processes in this way.

## Tool integration for automated software deployment

LANDESK Service Desk integrates with LANDESK Management Suite (LDMS) through LANDESK Process Manager (LPM) to provide automatic request fulfillment. The automation platform provided with Service Desk contains automation processes that respond to Requests at a specific status, and then select from the Request the relevant software package, the target computer and other supporting information, before passing that information to LDMS for automatic deployment. When the software has been deployed successfully, LPM passes the result to Service Desk and the Request is updated. The **Software Request** process included with Service Desk provides the basis for this integration.

**NOTE:** For more information about the **Software Request** process, see the *LANDESK Service Desk Prebuilt Content Guide*.

From a service-centric perspective you may also want to consider the Release of the Service itself. This transition into Operational status is driven by the Service lifecycle, but remember that you can modify the default queries in the **Service Catalog** component to restrict the Services that appear to be only those that are at a **Released** or **Live** status. This prevents users from requesting Services that are not yet available.

# Service Asset and Configuration Management

Configuration Management is the process responsible for maintaining information about Configuration Items required to deliver an IT Service, including their Relationships.

Much of the advanced ITIL Configuration Management activities in LANDESK Service Desk rely on linking your CIs into Structure diagrams, on versioning, and on Managed Versioning (managing CI data and versions from Change processes). To enable this behavior, you must have the LANDESK Configuration Manager component of Service Desk.

**NOTE:** For more information about using the Configuration Manager component, see the *LANDESK Service Desk Administrator Guide*.

## Naming CIs

When you design your CI windows and attributes, remember that the **CI Name** attribute must be unique. With IT equipment this is rarely an issue, but if you already provide asset-tagging of all IT equipment then this Tag or Asset ID could be a logical choice of attribute to use to define your CIs. However, remember that if you import this data, the unique identifier must also exist in the source of the data being imported.

**NOTE:** Do not use IP addresses as your unique name for a CI. Although IP addresses are unique, they are often dynamically allocated.

When creating CIs, either manually or from imports, the naming and format conventions set on the CI are enforced. By default this includes the unique CI Title. Duplicate CIs cannot be created in the CMDB. CIs are, by default set to have soft-deletion, therefore even if a CI is deleted a record of it (and its unique name) still persist in the database, this means that a new CI with the same name can still not be imported.

Typically, existing CIs are discovered using third party discovery tools and imported into Service Desk using the Data Import component. These act as your baseline for future management of the CIs through Change and Release Management. If you place any attributes under Managed Change Control, do not update them through importing after the baseline has been taken.

## Availability Status for CIs and Services

Service Desk provides a range of ways to track and present the current status of CIs and Services for both IT staff and customers. The simplest approach is to add a reference list to your CI window, providing a dropdown list of **Service Status** values (**Available**, **Impacted**, **Unavailable**, **Withdrawn**, and so on). You can then change these statuses manually when required.

Alternatively, you can change this value automatically from within other processes. You could use a Change process, for example, to set the status of a Service or CI automatically when it is created or resolved.

In both cases, you can design Report Templates in Object Designer to enable you to present attractive and informative Service Status displays in Web Access. For example, you could create a query that used a Report Template so that end-users can see a list of services with icons representing their current status, driven by the status value on the CI.

**NOTE:** See also Service Portfolio Management on page 35 for information about how you can link the Service Lifecycle status to a Service CI.

## Vital Business Services

---

You may choose to indicate that your Services or CIs have a vital business function. Add a boolean checkbox to the Service or CI window to indicate this. Consider also using the different relationship types in the CI structure component to highlight with a different color those CIs or Services that have a vital business function.

## Baseline

---

If you choose to follow ITIL guidelines and specify baseline versions of CIs, you can either add a dropdown list of **Version Type** to your CI, or enable version control and use the standard **Description** field to describe whether a version is a baseline.

## DSL and DML

---

Previously described in ITIL v2 as the Definitive Software Library, if you are following the ITIL v3 guidelines you can keep a track of your source master media within the Service Desk CMDB by defining a Definitive Media Library (DML). This is a grouping of CIs, with one for each definitive source. Typically these are grouped in a CI structure, although using a checkbox (in DML) and a simple query (DML) can also provide this view.

## Complex/Component CIs

---

You may want to build complex CIs that comprise a number of linked CIs. As with many areas in Service Desk, there are different ways to approach this, one of which is described below.

### To set up complex/component CIs:

1. Use the **CI Structure** component to link your CIs together with relationships. The links in the diagrams are real data relationships, and joining CIs in a diagram also joins them together at a data level.
2. Use the **CI Relationship Types** tree in the CI Structure component to create new relationship types for different types of linking. You can set CI versioning to be manual or automatic on any CI type. If you require both the components and the parent CI to be versioned, use **Managed Versioning**. This automatically sets a version number on a CI when the **Commit** action is used in a Change process.
3. To increment component version numbers, include two **Add Configuration Item** actions to your Change process, so that both the parent CI and the child are selected and managed. To do this when prompted by the Change process, click **Add Configuration Item** and select the parent CI. When then prompted to select the child by the process, click **Add Configuration Item** and select the child.

## Viewing affected Services from a CI window

---

You can view the related CIs and Services from a Configuration Item window by right-clicking the CI window then clicking **Impact Analysis**. This is possible only if the **Impact Analysis** option is made available on that window.

## Development Environments

---

If you are managing Development and Live environments in your CMDB, then there are two recommended options to track whether a CI is in **LIVE** or **DEV** usage:

- Add a **Usage** reference list to your CI window, and use this to select the environment used.
- Place CIs in groups representing Live and Development copies of Services. In the **CI Structure** component, drag the required CIs onto a structure diagram, and group the two collections of CIs by dragging around a group then clicking **Group Items** in the **Actions** list.

## Reporting on Configuration Items

---

You can report on various aspects of your CMDB using the reporting capabilities in LANDESK Service Desk. Some useful reports to run are:

**RFCs by CI** – build a query on the **Change Configuration Item** object and group it by **CI**. This will list all of the changes associated with the various CIs.

**Failing CIs** – build a query over the **CI** object and filter it by those at the unavailable status.

**All CIs** – build a query over the **CI** object and filter by those that are not soft-deleted.

# Service Level Management

An important part of the definition of ITIL Services is the understanding and agreeing of the correct levels of service for the customers of that service. Using a Service Lifecycle approach with LANDESK Service Desk enables you to guide the Service Level Manager through the correct steps as a function of defining and managing each individual Service.

To enable the requirements gathering, agreement creation, and targets setting required for mature Service-Centric service level management, your Portfolio process contains actions to:

- Review and agree Service Level Requirements (SLRs) and Service Level Targets (SLTs) with customers. Collections built during the Service Lifecycle are used to capture multiple SLRs and SLTs; clicking the actions Add Service Level Requirements enables you to capture this information.
- Create Service Level Agreements – with agreement scope, creation date, contacts and targets created as collections on the Service window. For each agreement, select the Customer, User or Third Party from the customer/user list. Once selected, click the hyperlink on the list to access the full Customer details. Users with the appropriate privileges can view and modify all of the Customer groups, the End-users, the Third Party groups and the Third Party Contacts using the **Administration** component.
- Produce an SLA document. Typically this is manually produced and attached to the Service CI.
- Review agreement content with the Service Level Manager to confirm all agreements are captured and stored, and updates have been applied in Service Desk – for example Response Levels and Escalations – to support the agreements. The Service-CI window displays fields that describe the typical broad levels of service provided, including the Service Hours.
- Build **Service Quality Plans** (SQP) (detailing all measurement methods and inputs into monitoring Service performance) – usually attached as documents attached to the Service Portfolio process.

You can access the complete Service Level information from the Service CI, and also from the **Service Level Management** component in **Administration**. In either case, if you have the appropriate privileges, you can modify all of these values and records – including SLA, OLA, UC and the SLTs described above. However, we recommend that you set a Service Review date to identify when you need to review the agreements, and then modify your SLA query to group by Date to show a schedule of future service agreement reviews.

To see more detail on the internal and external service providers, expand the **User Management** tree in the **Administration** component. To see a list of external providers, expand the **Suppliers** folder. To see internal providers, expand the **Support Group** folder.

The selected Service Provider is visible on the Service Window. To see a list of the providers of Services, add the **Service Provider** field to the Service Portfolio query.

**Design Idea:** Create a separate Change process linked to your Service CI to provide a full and complex service review activity.

**Design Idea:** A query of Service-CIs showing the collections of agreements on each Service enables you to quickly click through and view each service and its associated SLA, OLA and UCs without needing to drill down into each one.

If you are building your Service Agreements in this way, you could also import on a scheduled basis from the Service Agreement tables into the Response Level Matrix to keep your automatic setting of Response Levels on processes aligned with your Service Lifecycle changes.

You may not be able to measure all of your SLM requirements through Response Levels. Typical SLA documents confirm a number of dimensions including Service Uptime and Availability. You may need input from Availability Management and Event management to track these in Service Desk. Using reports to bring multiple inputs from other activities together when you are tracking performance against complex Service Level Agreements gives valuable information.

For example, standard Crystal Reports include one providing availability measures, and another providing Response Level performance. Run these separately or combine them as sub-reports in Crystal Reports with your Response Level reports to see a view of both availability and responsiveness, showing overall performance against Service Level targets. An example of this is the KPI Summary report (available from the LANDESK Community), where MTTR and MTBF metrics are combined with response level information.

**Design Idea:** You can monitor Service Level Agreements visually or in a report in a number of ways, in varying complexity, based on your implementation of Service Levels.

1. Monitoring SLA response performance against agreements by both Service and Customer is most easily delivered through the standard **Service Level Agreement Monitoring (SLAM)** report – SLAM Chart.
2. Combining multiple inputs, for example where availability needs to be considered alongside response times combined with other external factors may require initial a Service Level performance review in Service Desk where an **Ahead, On Track** and **Behind** value is set on each current agreement.
3. Remember the use of LANDESK MI as an aggregator of multiple feeds over time. MI can take input from Service Desk performance and external sources and present their daily metrics in graphical form.

When producing Service performance reports, as well as by Service and by Customer, the same information can be invaluable in assessing the performance of your third party suppliers, or services that are fully supplied externally, such as some cloud services. The same measuring of availability and events can track how reliable and performant these external services are.

**Best Practice:** It is easy to become focused on the internal IT activities to achieve agreed SLA performance, and to assume that if these agreements are being delivered upon then IT is always doing a 'good job'. Sometimes the level of performance and IT service may match the agreement, yet still fail to satisfy the customers and end users. This can happen over time as the business expectations and needs change faster than the agreements in place with IT. Use the Survey ability of Service Desk processes to monitor on-going Customer Satisfaction. The default Incident process includes Survey actions and objects to enable each end user to record their current satisfaction with IT. Also consider using the standard Complaints process to enable your customers to record a Complaint – or compliment – through Self Service.

In support of this activity it is also important to ensure that you regularly perform **Service Reviews**. This is implemented by way of a Task process on the Service Portfolio process, enabling the management of Service Reviews throughout the lifecycle of the Service. Proposed review dates are specified on the **Chartered** collection of the process and can be used to trigger the creation of the Task record.

# Service Portfolio Management

Service Portfolio Management is the process responsible for managing the complete set of Services that are managed by a Service Provider.

You can operate a service-centric approach to Service Management using LANDESK Service Desk. This is a key concept behind ITSM and ITIL 2011, and takes a typical ITSM service desk beyond the basics of Incident, Problem and Change management into the more advanced activities of Capacity Management, Availability Management and Financial Management.

## Service lifecycle

To start managing a portfolio of processes, firstly create your new Service Lifecycle process. This would typically step through all the activities in moving a Service through ITIL activities defined in Strategy, Design, Transition, Operation (with Improvement) and eventual decommissioning and end of life. In ITIL, Services in the Portfolio pass through these activities and become **Defined, Analyzed, Approved, and Chartered**, indicating the movement through the Lifecycle process. The activities in these ITIL phases enable you to plan and specify the future KPIs and usage of the Service. Your Portfolio of Services becomes the full set of lifecycle-driven Services that are at various stages in their Service lifecycles. To create a new portfolio record, raise a new instance of the Service Lifecycle process. This then guides your business through the required steps – for example creating a value proposition, a **service design package (SDP)**, capacity plan, availability plan, and defining future availability and capacity limits. Typically this includes linking the Lifecycle process to the Service CI so that the Status of the Lifecycle process can be seen from the CI. The first step of this process is to add an existing CI to the process, or to create a new CI. After this, you can add requirements, authorize, design, build, test, and then release the Service.

When the Service is released, the process moves to a **Live** status. At this status, you can add **Create Child Change** actions that create an instance of this Service, which can then be managed separately.

From the **Live** status, you also add a **Retire** action, to move the Service lifecycle to a **Retired** end status.

You can then add a filter query tab to the Service window to display the status of the Service Lifecycle process for the Service you are viewing.

Changing the default queries used in the Service Catalog to display only those Services that have a Lifecycle status of **Live** enables you to control the availability of Services through the Service Catalog to only Live services.

**IMPORTANT:** The ability to use the lifecycle process to enforce anticipated expected usage, capacity, availability, and cost values is the key to Service Centric ITSM. The Service Lifecycle guides IT into planning the expected values for operation, against which the actual values can be compared.

This leads to monitoring of service usage and performance against targets, and then onto service improvement as limits are exceeded.

Assessment of future changes, growth and risk all take place based on the current values, and target and maximum values on the Services in Service Desk. Often, proactive service improvement activities include reference to identifying risks, and implementing the changes to negate those risks before they bring current values beyond service value thresholds.

Other fields typically included on a Service include **Cost** (to IT) and **Charge** (to business). These directly support Financial Management activities and enable charge-back and financial service accounting.

**Design Idea:** Your Service CI can be created manually when prompted from the process, or can be created automatically populated from values recorded in activities on the Lifecycle process.

The status of the Lifecycle process defines the status of the Service. You can see this status from the Service by adding a filter query to the tabs on the Service CI window showing the Service Lifecycle status. You can also add any other attributes that you require on the CI (name, purpose, description, cost code, business impact, and so on) to the Service window.

**To view your Service Portfolio:** query a list of Services lifecycles with status, name, description and any other fields required.

**To view your service pipeline:** query a list of Services as above whose statuses are all in pre-Operation phases.

**To view your Service Catalog:** query a list of Services as above with a status of **Live** (or other suitable status in the Operation phase).

**To view your retired services:** query a list of Services as above with a status post-live (decommissioning or retired).

To view your services in use: query over the User Config Item table to view services by customer or customers by service.

Remember that your Service, once in live operation, is ideally under change control. Any developments and evolution to that Service are managed as standard Change processes, linked to the Service CI. Also once in operation you can update the Service window (or a collection) with values obtained from external systems using a scheduled Import. This provides you with a location to store snapshots of the latest monitored availability/capacity information to compare against those figures identified in the planning stages. Typical metrics you can derive and compare against planned limits includes Service Usage (the number of users subscribed or linked to the service).

**NOTE:** You can deliver Services from a number of routes. Once you have the structure enabled as described, you can include Services delivered to you from third parties, which may or may not be component Services in larger IT Services. The same concepts apply. The supplier of the Service, internal or third party, is described in the **Service Provider** field on the Service window.

## Continual Service Improvement

---

**Design idea:** Continual Service Improvement can be a looping activity of performance review on a live service. You can enable on-going **service improvement plan** (SIP) activities by setting a **Due Date** for review and a status for **In Review** or **Out of Review**.

As you refine the definition of your Service, the Service CI window becomes an increasingly critical point to hold all information relating to the Service. When you publish Services through the Service Catalog, the Service Description is usually presented to the end-users, so remember to word the descriptions in user-friendly text. If you need a formal document describing the service, you can either attach a separately prepared document to the Service, or you can generate the document automatically from the values held on the Service. For example, you can run a Crystal Report that formats into a document layout, or you can automatically generate Word documents attached to the process at the required point in the process.

**NOTE:** For more information about Word Integration, see the *LANDESK Service Desk Administrator Guide*.

## Self Service and Service Catalog

---

When a service in the portfolio reaches a status of **To release**, it appears in the **Service Catalog Administration** component to be published.

You can change the appearance of Self Service and the Service Catalog. The Service Catalog layout is defined in Service Desk using a **Reports Template** on the Configuration Item object in Object Designer. By editing this, you can change the appearance and layout of your catalog. For example, you can add other fields such as: Cost, Approval, Supporting Documents, and so on. Although you can include usage documentation at this point, you may prefer to provide this as a part of the fulfillment process. Refer to the LANDESK Community for examples of different Self Service and Catalog designs.

## Managing Multiple Services

---

There are several ways to manage multiple linked Services. Remember that Services are CIs, so you can link them in CI structures to all component CIs and Services. This provides a graphical Service relationship view as described elsewhere in this document. However, in offering the ability to request new Services through the Catalog you can also link delivery of items in more than one way:

- On the Fulfillment process raised from the Catalog, you can include the steps to ensure that each component delivery is made. For example, assign the Request to Education to provide Training, or generate a new child Request for Training.
- The Bundle function in the Catalog enables multiple linked Services to be bundled together, so that one Request initiates a bundle of individual Requests.
- Using the Shopping Cart, you can request multiple items. The Cart will not prompt if a required sub-item is not included, so this option provides more choice, but less enforcement.

## Defining Service Levels

---

Set the definition and agreement of the levels of service available with your new Service during the Design stages of your Portfolio process. Usually, standards such as Gold, Silver, and Bronze are defined at early stages, and then linked to the individual customers as and when the customer-service agreement takes place.

Define your Service Level standards on the CI Service window (using text fields labelled **Gold**, **Silver**, **Bronze**, and so on). You can also attach complete SLA documents using the **Attachment** action or button. You can then link to the relevant SLAs for the Service, either by clicking the Service CI on the Lifecycle definition, or clicking the attached documents collection.

As mentioned above, you can link from your Service definition to Availability, Capacity and Financial management activities. The target values identified on the Service for availability and capacity are set in the design stages, and the latest values are updated automatically from external monitoring tools into collections on the Service definition. Click the **Capacity data** and **Availability data** tabs to see Availability and Capacity 'latest' values. These values are also used to produce Capacity and Availability planning and performance reports. For financial management, click into the target values for **Service Cost** and **Service Price**.

# Working with External Data Sources for Advanced ITIL Activities

This section describes how to use external data sources to support certain advanced ITIL activities.

## Connection to external monitoring tools

Use the Import Source and Import Mapping settings in the **Data Import** component to connect to one or more external monitoring tools. Then you can map the imported fields to the **Availability** and **Capacity** collections on the Service and import this data using a daily schedule using the Scheduler component.

**NOTE:** Service Desk enables you to connect to many generic sources of data (SQL Server, Oracle, Access, Excel, Active Directory, LDAP, and so on). External sources of data imported as described may be monitoring and tracking tools, or business and management tools, even down to individual spreadsheets. Your various data sources may provide alternative APIs, but if you are using the above supported import connections, no custom configuration is required. Service Desk also supports connections to database views, and many monitoring tools provide database views pre-built to translate complex table contents into a simple presentation of key data such as **Uptime, Downtime, Cost, Average Response Time, Storage usage, average processor usage, Department Budget**, and so on.

When this data has been imported, it is also available for presentation and distribution through all of Service Desk's reporting and information routes, including dashboard gadgets, e-mails and reports.

**Design Idea:** It is not only imported data that can be presented through Service Desk. You can use the same Data Connections set up for import to run live queries on external data sources, and those live results can be displayed on dashboard, e-mails, RSS feeds and so on.

## LANDESK Management Information

LANDESK Management Information (MI) provides a further benefit in this area. MI can connect to external sources of data (such as Availability and Capacity monitoring tools) and run calculations on those sources, storing the results of those calculations in warehouse tables every day. Then MI's Trend chart gadgets enable you to display multiple trends of captured values over time against defined tolerance and threshold (breach) values. In this way you can present charts tracking, for example, CI or Service availability or capacity over time against a defined target.

MI also connects to Service Desk data. If you are running Event Manager to track device and service availability, you can create a metric showing the number of Events received, and display this in a chart over time in MI to see a view of Reliability (or unreliability) trends.

**NOTE:** See also the *LANDESK Service Desk Management Information Guide*.

# The Design Ideas Database

The Design Ideas database is a database with many ITIL processes pre-built inside it for you to use as a template for your own designs. Below is a description of how each of the processes is managed and followed in the database.

## Incident Management

The Incident process is described by ITIL as being the process responsible for restoring normal service as quickly as possible while minimizing adverse effect on business operation. Therefore, the process is designed to help the analyst who owns the incident collect and manage information pertaining to the incident and use it to find a solution.

In order to capture information about the Incident being logged to assist with diagnosis, resolution and reporting, the most useful fields to include are:

Full Name: Max Lewis	Company: Our Company
Login ID: Max.Lewis	Department: Exec
Email Address: Max.Lewis@ld.local	Phone: 100-1015
Call Back Method:	
<b>Incident Details</b>	
Details: My laptop doesn't work the way I expect	
Summary: I need help	
Category: Hardware - Laptop	Resolve on Creation? false
Impact: Medium	Priority: 2 Day Fix
Urgency: Medium	Status: Open
Ignore Agreements? <input type="checkbox"/>	
<b>Configuration Item Details</b>	
CI Type: Workstation	Config Item: WIN7-DEMO
<b>Current Assignment Details</b>	
Analyst:	Group:
<b>Status Info</b>	
Breaches in: Minutes	Unresolved: false
Assign Count:	Major Incident: false
Created: 31 July 2014 16:14:25	By: SA
Updated: 31 July 2014 16:14:25	By: SA
	Reopened: false
	Clock Stopped: <input type="checkbox"/>

- Creation Datetime (default)
- Update Datetime (default)
- Creation User (default)
- Update User (default)
- Affected User (Raise User, default)
- The Incident source (how the Incident was logged)
- Contact information of affected user (telephone number, email address – default)
- Preferred call back method (add a reference list for this)
- Category
- Priority (calculated from Impact & Urgency and/or escalation breaches on SLA)
- Impact
- Urgency
- SLA
- Symptoms (description – default)

- Diagnostic activities (this is the notes collection – default; or a custom new collection – see Designer)
- Status (default)
- Known errors associated with this Incident (by category, CI, CI type or keyword)
- CI (default)
- CI type (default)
- “Caused by change” – action to associate the underlying change to the records it causes

## Process Actions

**Create Request.** This action allows the analyst to create a request from the incident. This is usually done when it becomes apparent that the reason the user contacted the Service Desk is not because of a service outage but because they require a service or some information.

**Assign to user/group.** This action is used alter the assignment of an incident from one group, role or user to a different set of users. There are several ways of doing this in the process designed in the database, you should use the method most appropriate to your organization and the situation with the incident. For example, if you are a distributed organization with teams of service desk analysts that support users in different languages then the ‘Assign to Local Support’ assignment action is the most appropriate.

The assignment window contains fields for selecting the appropriate group, role or user and a mandatory title field. The title and description are used in the notification to the group or role members or the individual to tell them why the incident has been assigned to them.

**Add affected user.** This action is used when a major incident has been identified and other users are notifying the service desk that they have the same issue. This action will automatically add an incident for the affected user and attach it to the major incident. When the major incident is resolved, all the attached incidents will also be resolved and the affected users notified that the issue has been resolved

**Close.** The close action is used when the assigned analyst has confirmed with the user who reported the incident that it has been resolved to their satisfaction. A close category is used for reporting purposes to understand what the incident was ultimately caused by.

**Resolve.** The resolve action is used to identify what was done to resolve the incident as reported. This information is often used to create knowledge from for use in future incident investigations. A 'Caused by Change' field is included in the resolution to identify if a particular change was the cause of the service outage that led to this incident being reported.

**Complete survey.** The complete survey action can be published to the users or analysts and is a way for the user to report their experience and satisfaction with their contact with the service desk. You can make surveys only required for certain users, or on every x incidents logged.

**Change priority.** This action is used to ensure that the priority can only be changed by authorized users (using privileges) and requires that they complete a justification field to audit why the change in priority was made.

**Add Problem.** This is used to select an existing problem record to associate it with the current incident to show that the root cause of the incident is still being investigated.

**Create Problem.** When an incident cannot be resolved or the root cause of it is not identified during investigation then a problem may be logged to deal with managing the underlying cause going forward. This action is used to create a problem record and associate it with the incident.

**Create Request for Change.** On occasion, an incident that has been logged can only be addressed by making a change to the environment or turns out to actually be an RFC. This action creates a change record from the incident and then associates the records to each other.

**Raise Change**

Change Summary:

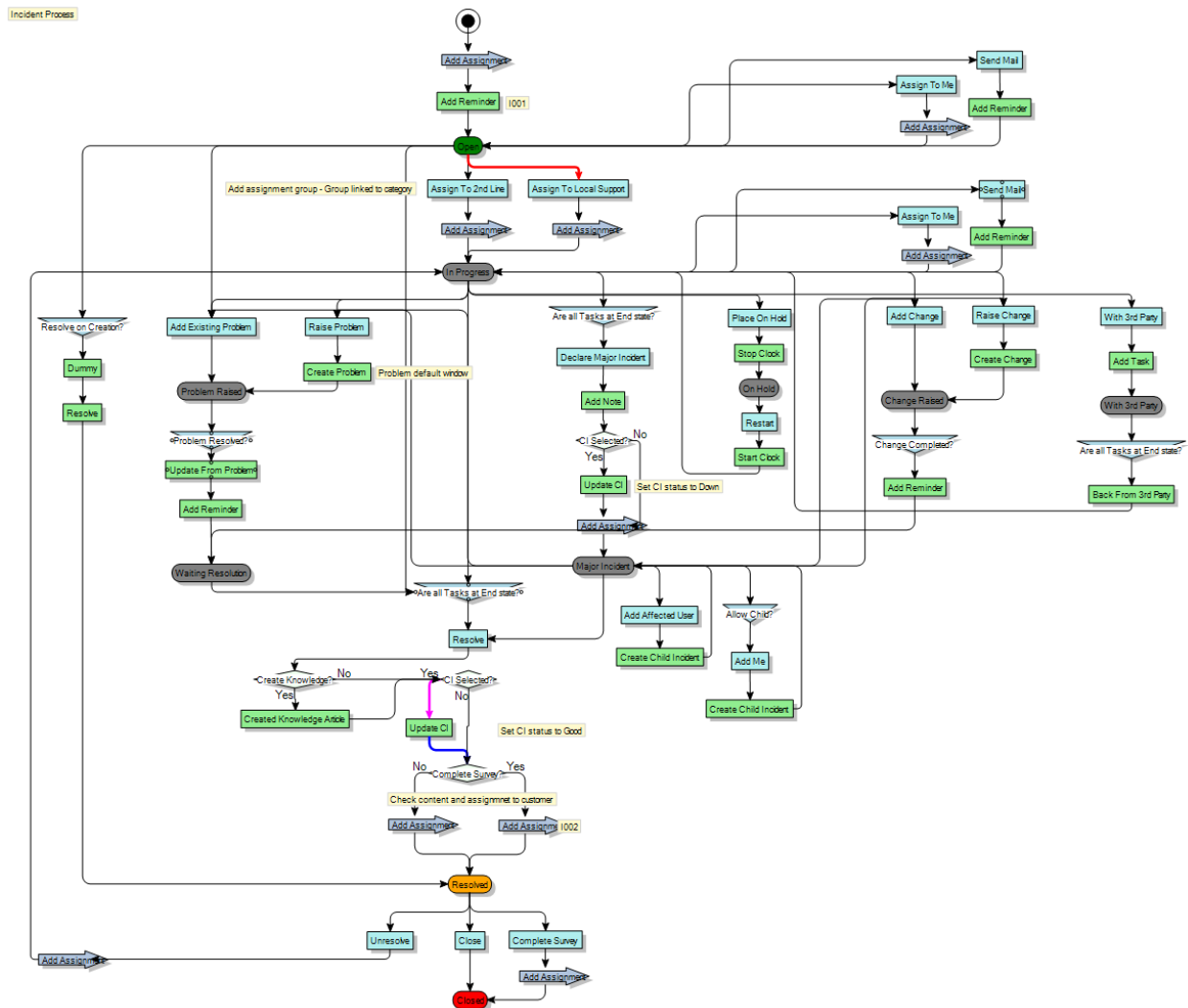
More Details:

Change Category:

Severity:  Priority:

Created On: 31 July 2014 16:26:47 Created By: SA

## Process Design



## Problem Management

A problem is logged either reactively, in response to a particular incident or group of related incidents or proactively when a trend in categorization is identified.

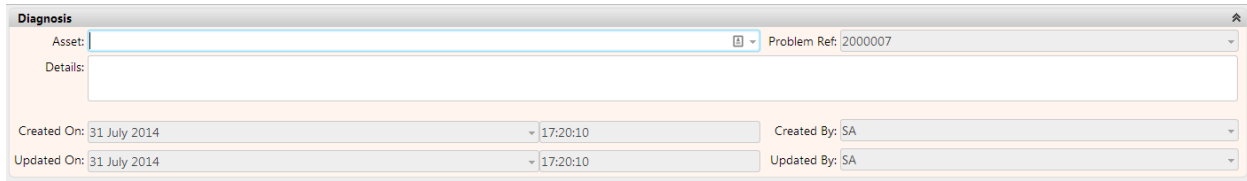
In order to capture useful information about the Problem being logged to assist with diagnosis, resolution and reporting, the most useful fields to include are shown on the screenshot of the window below and described in detail.

Raise User	<input type="text"/>	Problem Source	<input type="text"/>
Phone	<input type="text"/>	Email Address	<input type="text"/>
<b>Problem Details</b>			
Title	<input type="text"/>		
Description	<input type="text"/>		
Category	<input type="text"/>		
Problem Impact	<input type="text"/>	Response Level	<input type="text"/>
Problem Urgency	<input type="text"/>	Priority	<input type="text"/>
<b>Configuration Item</b>			
Configuration Item	<input type="text"/>		
<b>Status Information</b>			
Status	<input type="text" value="Open"/>	Lifecycle	<input type="text" value="Problem"/>
<b>Creation Details</b>			
Created By	<input type="text" value="SA"/>	Creation Date	<input type="text" value="31 July 2014"/>

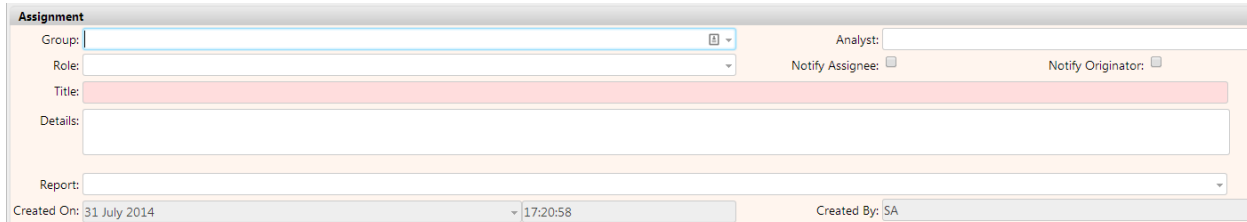
- Creation Datetime (default)
- Update Datetime (default)
- Creation User (default)
- Update User (default)
- Affected User (Raise User, default)
- The Problem Source – this may be an Incident or Event number
- Contact information of affected user (telephone number, email address – default)
- Preferred call back method (add a reference list for this)
- Category
- Priority (calculated from Impact & Urgency and/or escalation breaches on SLA)
- Impact
- Urgency
- SLA
- Symptoms (description – default)
- Diagnostic activities (this is the notes collection – default; or a custom new collection – see Designer)
- Status (default)
- Known errors associated with this Incident (by category, CI, CI type or keyword)
- CI (default)
- CI type (default)
- “Caused by change” – action to associate the underlying change to the records it causes

## Actions available on the process

**Add Diagnosis.** The first action available on the problem process is used to add details of any diagnoses of the problem record.



**Add Assignment.** This action is used to pass responsibility for the problem to a different group or person.



**Add Attachment.** Use this action to associate any supporting files with the problem record.

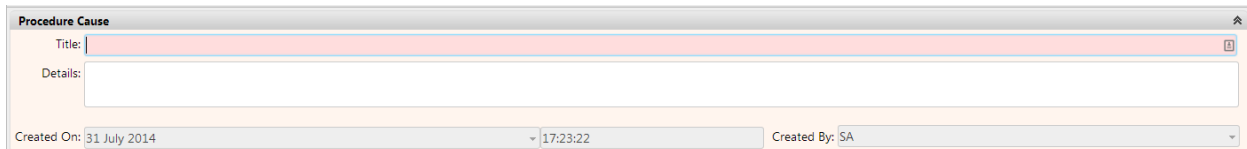
**Add Incident.** Use this action to associate the problem record with incidents that already exist in the system.

**Add Note.** The notes of a problem may detail information on the diagnostic steps already taken and can be used as an audit of what has happened so far.

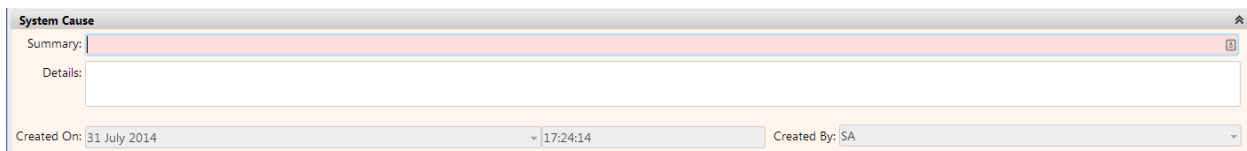
**Create Request.** This action is used to create a request record from the problem record.

**Detach Incident.** Occasionally an incident may be attached to a problem that further investigation reveals to be an incorrect linkage. Use this action to remove the association between the incident and problem records.

**Add Procedure Cause.** This action is used to add details of what the cause of the problem was if it is identified as a procedural issue.



**Add System Cause.** This action is used to add details of what the cause of the problem was if it is identified as a problem caused by the system. This will often result in a change request being raised to rectify the underlying issue.



**Create Known Error.** A known error is created when a system cause is identified and a workaround has been identified. This is then published to the knowledge base for use when investigating incidents.

**Known Error**

Known Error Type:  Config Item:

Summary:

Error Details:

Workaround / Solution:

Linked Problem: 2000007

Created On: 31 July 2014 16:25:55 Created By: SA

Create Knowledge?

**No Known Error.** Use this action when no known error can be created for the problem.

**Request Change.** If a problem has a system cause then an RFC should be logged to have an alteration made to the environment in order to resolve the root cause. The details of what needs to be changed and why should be identified on this action. A change category, severity and type should also be completed.

**Request Change**

Change Summary:

More Details:

Change Category:

Severity:  Change Type:

Created On: 31 July 2014 16:31:29 Created By: SA

**Resolve.** When a problem has had the root cause addressed it can be resolved, details of the resolution should be logged and a category for the resolution added. If the problem requires a major problem review, it should be noted as part of this action.

**Resolution**

Category:

Resolution:

Details:

Resolved On: 31 July 2014 17:32:19 Resolved By: SA

**Add Change.** Sometimes the root cause of a problem may identify a change that needs to be made that is already logged and pending approval or implementation. Use this action to associate the problem to an existing change record.

**Major Problem Review.** This action is used to record the details of the major problem review, including any details of the lessons learnt and the recommendations for alterations to the processes or systems in the future.

**Close.** The close action is used when a problem has been successfully completed. In addition to any further details, the closure category should be included to help identify and patterns.

**Closure**

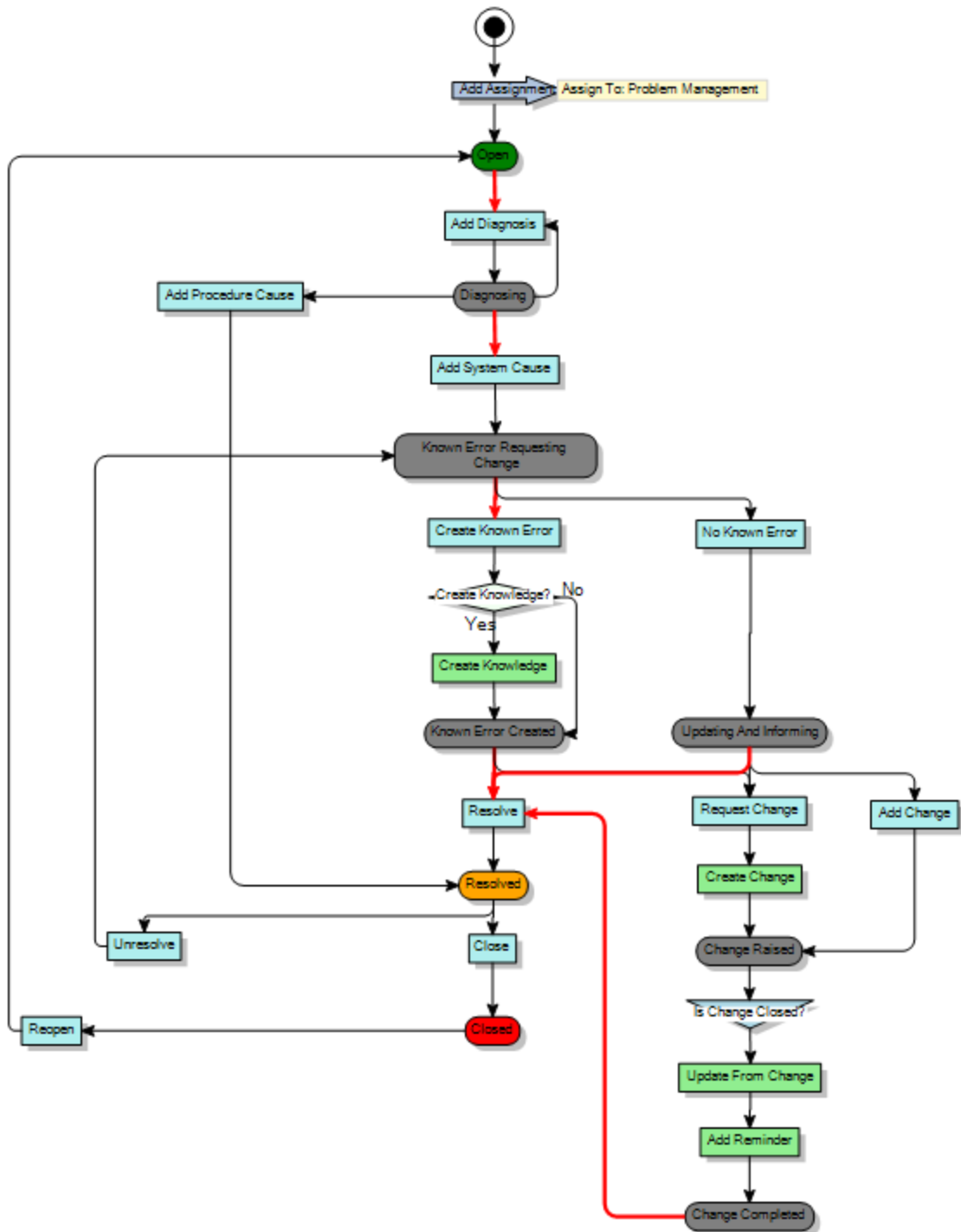
Category:

Title:

Details:

Created On: 31 July 2014 17:33:42 Created By: SA

## Process Design



## Change Management

Change Management is used to manage alterations to the systems in the IT environment while ensuring minimum disruption to the IT services.

In order to capture useful information about the change being logged to assist with deciding if to approve and when to schedule it the following information should be captured.

User Details	
User:	<input type="text"/> Dept: <input type="text"/>
Name:	<input type="text"/>
Phone:	<input type="text"/>
Change Details	
Details:	<input type="text"/>
Summary:	<input type="text"/>
Category:	<input type="text"/>
Affected Service:	<input type="text"/> Change Type: <input type="text"/>
Service Level and Status Details	
Impact:	Priority: <input type="text"/>
Urgency:	Outside Agreement? <input type="checkbox"/>
Lifecycle: Change	Status: Open
Clock Stopped:	Is Breached? <input type="checkbox"/>
Creation Details	
Created: 31 July 2014	17:40:18 By: SA
Updated: 31 July 2014	17:40:18 By: SA

- Raise User (or source – Incident/Problem/Request)
- Their contact details
- Details of the Change to be raised
- Summary
- Category – this should include Major, Minor and Significant
- Affected Service
- Change type – these should be Normal, Standard & Emergency
- Impact
- Urgency
- Priority
- Status
- Lifecycle
- Created By
- Created Datetime
- Updated By
- Updated Datetime

### Actions available on the process

**Add Configuration Item(s)**. This action is used to select the CIs that will be changed by the change.

**Add Attachment.** This action can be used to include any supporting files or other documentation with the change record. This can include the business case, or risk and requirement documentation.

**Add Incident.** If the change pertains to an incident in some way, use this action to associate the records with each other.

**Add Note.** The notes of a change are used to keep a record of any further information related to the change record.

**Add Parent Change.** If the change record is one of several changes in a larger change project then use this action to associate the related changes together.

**Add Problem.** This action is used to associate the change record with an existing problem record.

**Create Knowledge.** Use this action to create knowledge information about the change record to assist users with understanding why something works the way it does.

**Article**

Title:

Type:  Configuration Item:

ISO 20000:  SOX Compliance:

FAQ Type:  Expiry Date:

Description:

Solution:

Analyst Solution:

Reason:

Relevant To:

Tags:

URL Link:

Effectiveness:   Knowledge Domain:

Created On: 31 July 2014 18:58:29 Created By: SA

**Submit.** This action is used to submit the change to the review process. If it is of type 'Emergency' then it will be presented to the eCAB, if it is 'Normal' then it will be reviewed at the next CAB meeting and if it is 'Standard' then it will miss authorization completely.

**Add Backout plan.** This action is used to add a Backout plan to the RFC as information that may be used when the change is being reviewed. This action is mandatory if the change is classified as being 'Major'.

**Backout Plan**

Summary:

Details:

Document:  No file chosen

Testing State:

Create Date: 1 August 2014 12:42:30 Create User: SA

**Add Change Proposal.** If the change is categorized as being a major change then it cannot be submitted for authorization until it has had a change proposal added to it.

**Reject.** This action is used by the change manager or CAB representative to reject an RFC. The window requires a categorization of the rejection as well as details explaining why it is being rejected.

**Rejection**

Category:

Details:

Created On: 1 August 2014 12:48:45 Created By: SA

Updated On: 1 August 2014 12:48:45 Updated By: SA

**Authorize.** If the CAB or eCAB is prepared to allow the change to go ahead then they should use this action to authorize it. The window includes the ability to categorize the authorization and add details of any further information.

**Authorisation**

Category:

Title:

Details:

Created: 1 August 2014 12:54:38 By: SA

Updated: 1 August 2014 12:54:38 By: SA

**Schedule and Assign.** Once a change is authorized it needs to be scheduled in to the change calendar and assigned to a group or individual to carry out the change. These details are captured as part of this action .

**Schedule Details**

Affected Service:  Schedule Date:

Assigned Analyst:  Start Time:

End Time:

Notes:

Created On: 1 August 2014 11:57:10 Created By: SA

**Implementation Failed.** If the change was not successfully implemented in the scheduled time then it should be recorded as a failure. Details of what went wrong should be captured.

**Details**

Implementation Failures:

Details:

**Status**

Create Date: 1 August 2014 11:58:45 Create User: SA

Update Date: 1 August 2014 11:58:45 Update User: SA

**Success.** This action is used to audit the fact that the change has been completed successfully.

**Commit Configuration Item(s).** Use this action to commit the changes to configuration items associated with the change record to the CMDB.

**Close.** Use this action to categorize the closure of the change and include any details from the change review.

**Closure**

Category:

Summary:

Details:

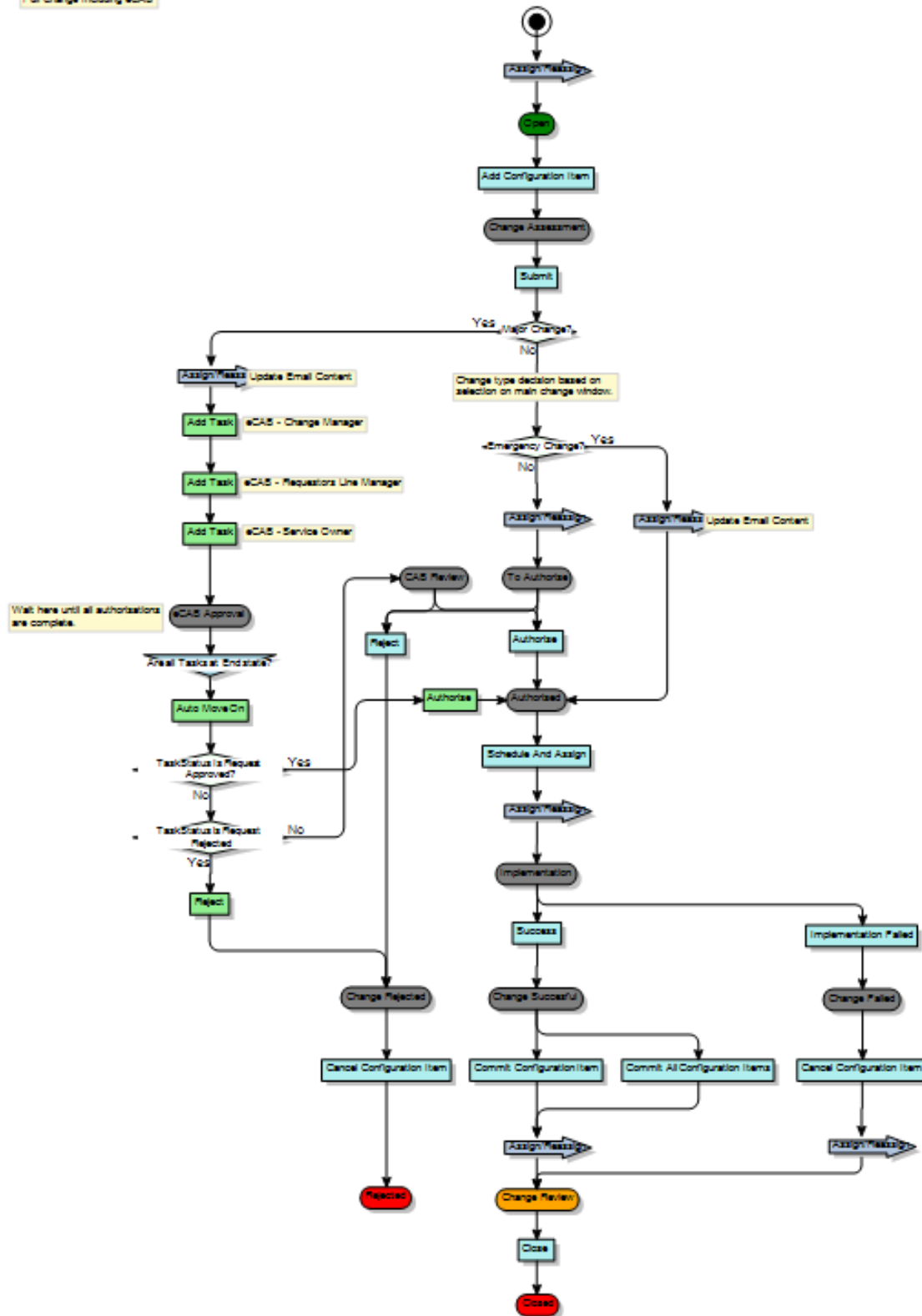
**Status**

Created: 1 August 2014 13:04:33 By: SA

Updated: 1 August 2014 13:04:33 By: SA

# Process Design

Full Change Including eCAS



## Request Fulfilment

The ITIL definition of request fulfilment is the process by which IT manages the lifecycle of service requests from their users.

Depending on the request to be fulfilled there is information that should be captured. Below is a description of what is captured for a generic request.

The screenshot shows a form with the following sections and fields:

- Requestor Details:** Requested For (dropdown), Requested By (dropdown), Name (text), Phone (text).
- Service Required:** Request Type (dropdown), Service Item (dropdown).
- Destination (If Request Applies to an Existing CI or Service):** Type (dropdown), Asset (dropdown).
- Request Details:** Summary (text), Justification (text), Impact (dropdown), Priority (dropdown), Urgency (dropdown), Status (Open, dropdown), Service Level Required (dropdown), Ignore Agreements (checkbox).
- Creation Details:** Created: 1 August 2014 13:21:29, By: SA, Update: 1 August 2014 13:21:29, By: SA, Is Parent a Bundle Process? (false, dropdown).

The fields that should be included on the window are:

- The requestor's name
- The requestor's contact details
- The service-CI being requested
- The CI to apply the service to (where appropriate)
- A summary of what is being requested
- A justification of why the user needs the service
- The Impact, Urgency and Priority of the request
- Details of who created and updated the request and when

## Actions available on the process

**Add Note.** This action is used to track any extra information about the request that isn't logged on the main window or as part of any of the other actions required to progress through the process.

The screenshot shows a form with the following fields:

- Raise User (dropdown)
- Category (dropdown)
- Title (text)
- Details (text)
- Created On: 1 August 2014 13:41:44, Created By: SA
- Updated On: 1 August 2014 13:41:44, Updated By: SA

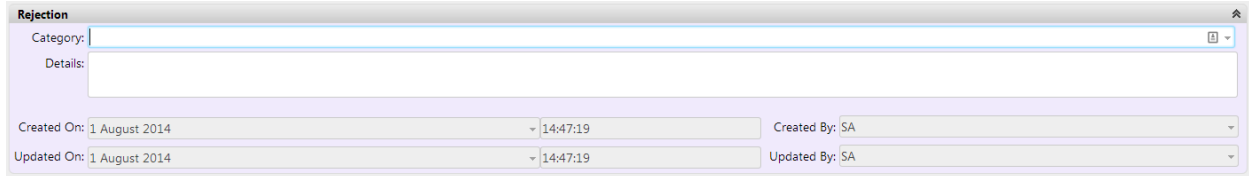
**Attach Incident.** Use this action to attach an existing incident to the request if they are related. Usually this will occur when an incident turns out to be a request.

**Create Incident.** Use this action to create an associated incident record.

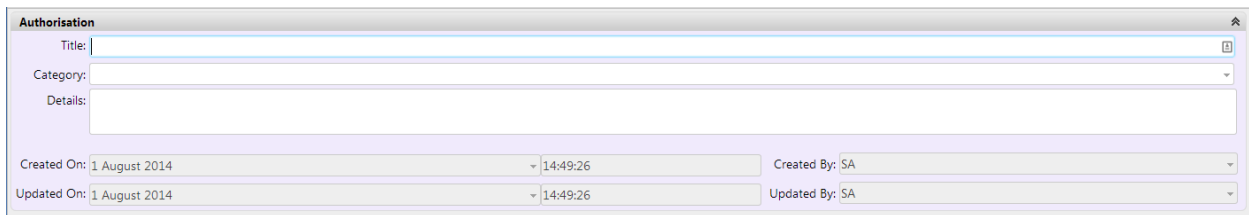
**Create Problem.** Use this action to create an associated problem record.

**Create Change.** Use this action to associate the request with a new RFC.

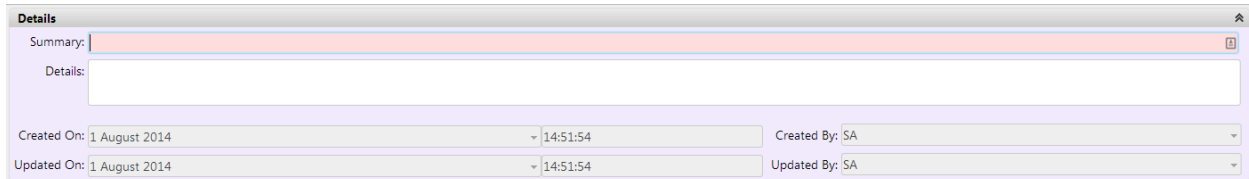
**Reject.** The process allows for several levels of authorization, including line manager, service manager, business and financial approval. If any of these parties want to reject the request they must use this action and provide a reason for the rejection as well as a categorization.



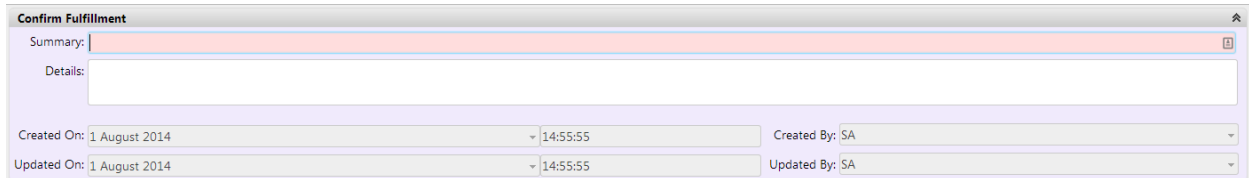
**Authorize.** If the authorizer is happy to authorize they must complete this step and both categorize and provide details of the authorization.



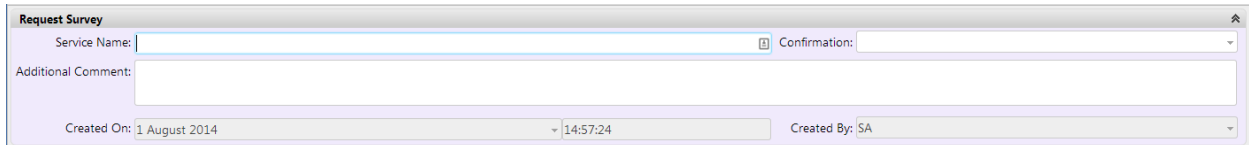
**Cancel Request.** This action is available for the request to be cancelled even after it has been authorized.



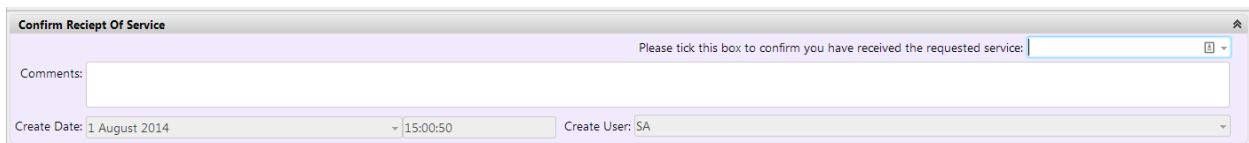
**Fulfilled.** This is used when the service has been fulfilled by IT, either by process automation or manually. The executor of this action should provide any extra details about this service.



**Complete Request Survey.** This action is used to allow the service requestor to provide feedback on their experience with the process.



**Confirm.** This action is used to confirm that the request has been fulfilled.



**Raise Ticket.** This action can be used by a user of a service to log an incident with the service desk about the service in question.

**Unsubscribe.** This action is used to remove the service from a user.

**Remove Service**

Summary:

Details:

Created On: 1 August 2014 16:13:14 Created By: SA

Updated On: 1 August 2014 16:13:14 Updated By: SA

**Removal Failed.** This action can be used when the removal of a service from a user was unsuccessful. This should include details of what went wrong.

**Removal Failure**

Summary:

Details:

Created On: 1 August 2014 16:14:36 Created By: SA

Updated On: 1 August 2014 16:14:36 Updated By: SA

**Confirm Removal.** This action confirms that the service was successfully removed from the user.

**Confirm Removal**

Summary:

Details:

Created On: 1 August 2014 16:15:54 Created By: SA

Updated On: 1 August 2014 16:15:54 Updated By: SA

**Reopen.** This action can be performed to reopen the request. The user who performs the action must give a reason for it to be re-opened.

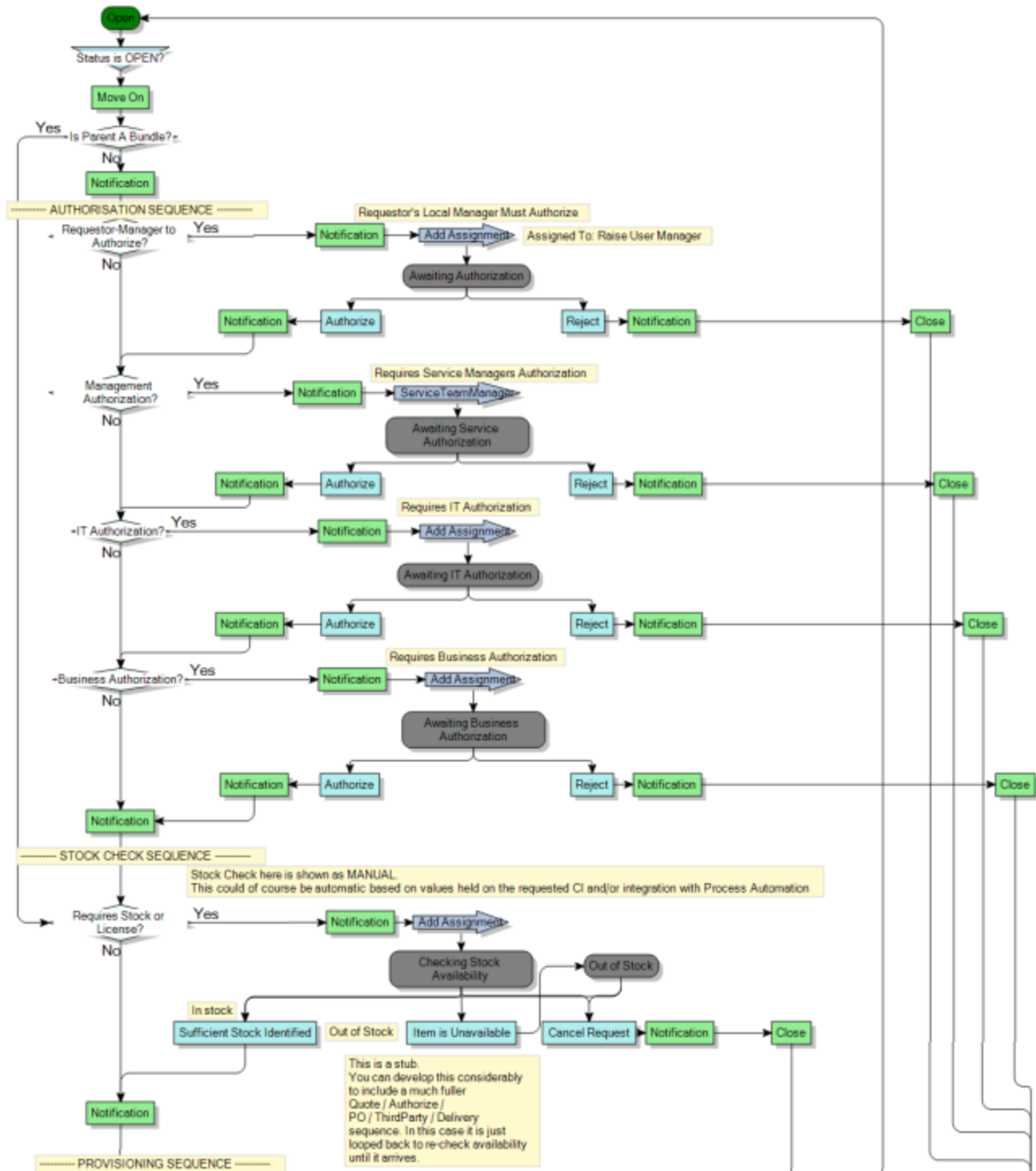
**Reopen**

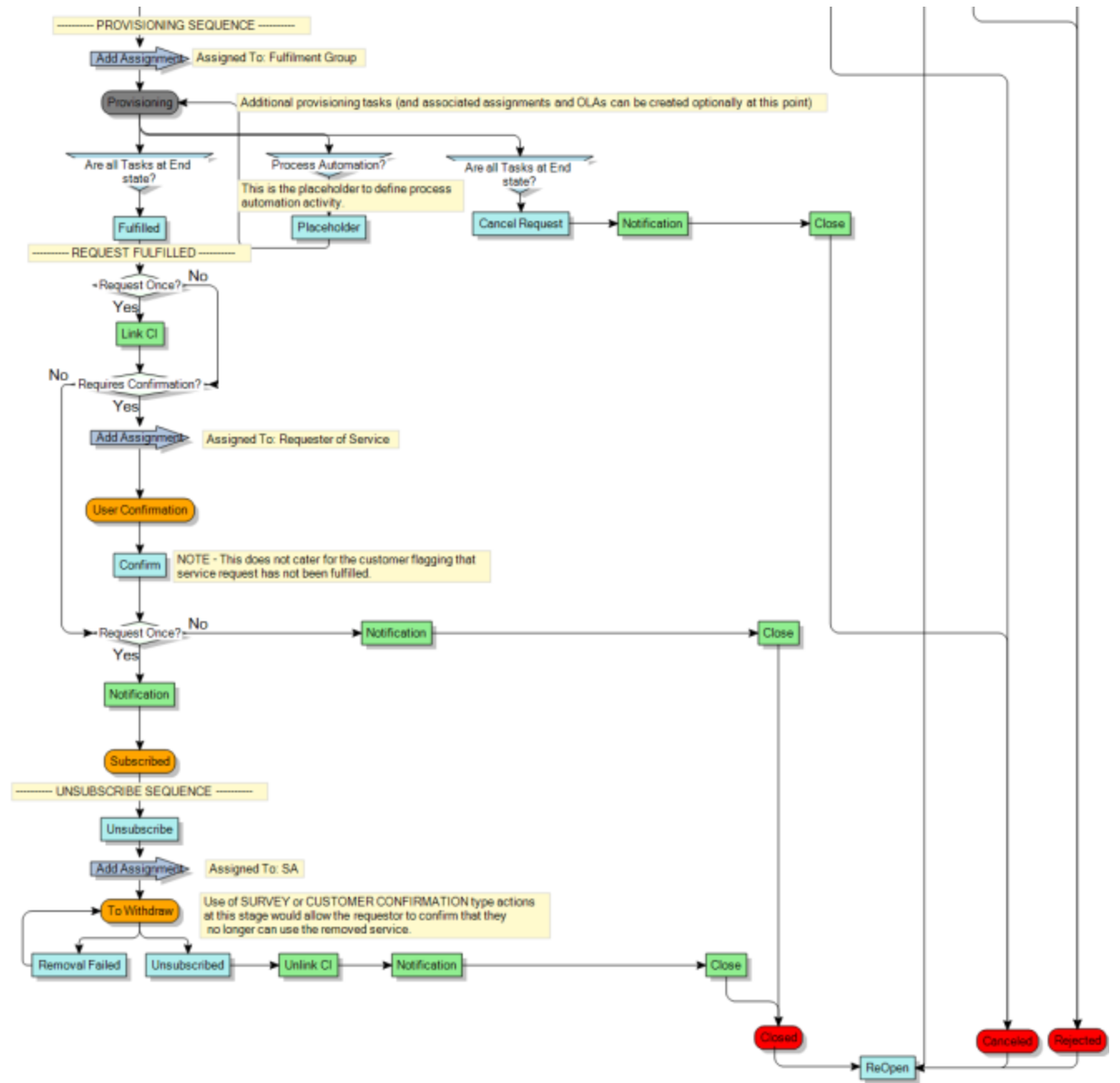
Reason:

Details:

Reopened On: 1 August 2014 16:17:06 Reopened By: SA

## Process Design





## Event Management

Events are usually detected automatically by appropriate monitoring software and logged in Service Desk if they meet requirements to require further investigation.

The details captured about an event should include the following items.


- The source of the event (usually the monitoring tool)
- The affected configuration item
- Any extra details about the event that the monitoring tool is able to provide
- The Impact and Urgency of the Event
- The priority of the event – as calculated from the impact and Urgency values
- The category of the event
- The type of event, this should be one of 'failure', 'information' or 'warning'

## Actions available on the process

**Add Note.** This action is used by the monitoring tool to add further information about the event as it progresses, it is used to report the status of the device and until the device is reported as being "up" (operational) the process cannot move forward.

**Resolve.** Once the event is over, the process can be resolved. The resolver must categorize the resolution, provide any details of what was done and specify a change that caused the event if appropriate.

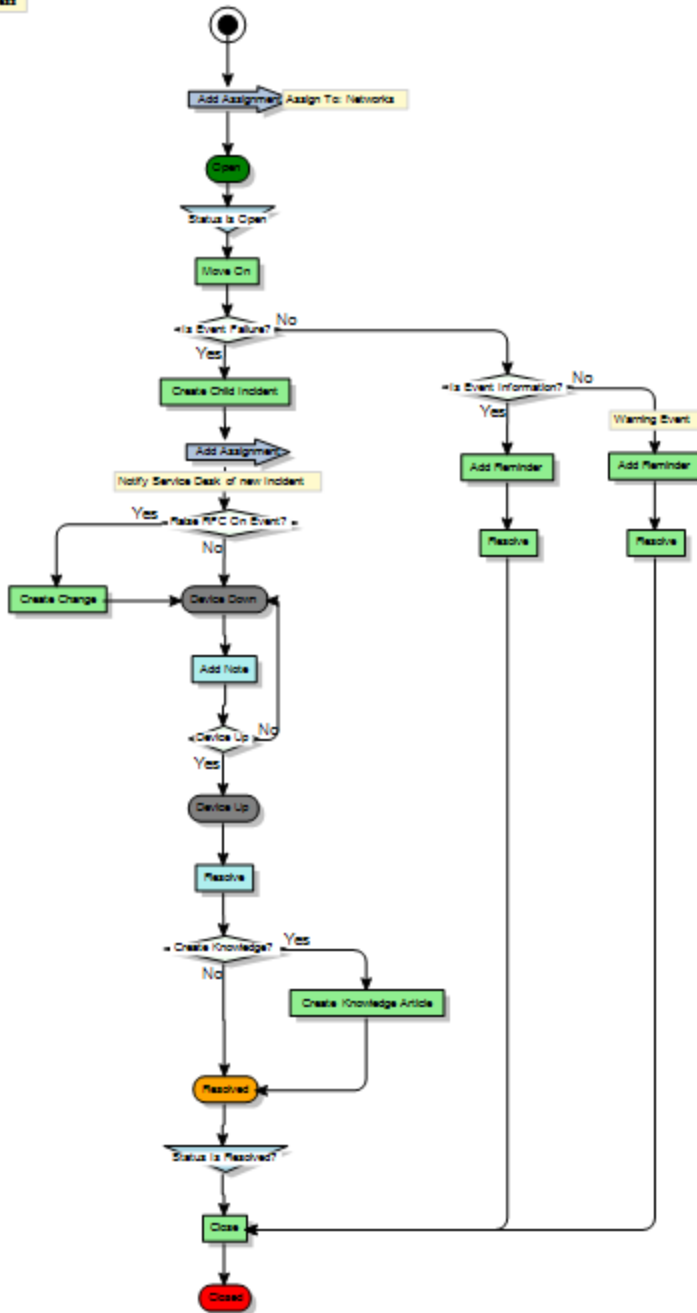
Having resolved the event, it is automatically closed. The details of these are seen in the audit history of the event

**History** 

- Close  
1  
1 August 2014 17:02:51 - SA
- Resolve  
1  
1 August 2014 17:02:51 - SA
- Add Note  
1  
1 August 2014 16:58:22 - SA

## Process Design

Event Process



## Service Portfolio Management

Service Portfolio Management is the over-arching process that manages Services from conception to retirement, including the design, testing, release, and decommission. As this process is central to so much of what IT does there's lots of data that should be captured, including the fields described below.

**Service Portfolio**

Service Name:\*

Description:\*

Service Owner:\*  Service Provider:\*

Business Users:\*

Processes Supported:\*

Provider Responsibility:\*  Customer Responsibility:\*

Service Live Date:  Raised By:

[Linked Service Cl:](#)  Lifecycle Status:

**Service Design**

Authorisation?  Financial?

Service Level?  Service Reporting?

**Information**

Created On:  Created By:

Last Updated On:  Last Updated By:

- Every Service must have a unique name
- Use the description field to describe exactly what the service is
- Service Owner
- Service Provider (if it's a third part service)
- Business Users
- Provider responsibility (what IT is going to deliver)
- Customer responsibility (what users will be expected to do with this service and how they can report any issues)

On the associated Service record you may want to record some further information pertaining to the service itself, not its lifecycle.

- Cost (to IT)
- Charge (to the business)
- The various service level packages available
- The various time zones and locations this service will/won't be supported in



## Configuration Item and Asset Management

---

Configuration Items are anything in the infrastructure that need to be managed for optimal business service continuity. As a result, any organization will hold data about many different types of CI, from servers and switches to software and services. Capturing the correct details about individual CIs is obviously done on a type by type basis.

Some fields common to most hardware-type CIs include:

- Name
- Display name
- Manufacturer
- Model (make this a string so that it can contain both letters and numbers)
- Serial number
- Asset Tag
- CPU
- Memory
- IP Address
- Hard disk size

Fields that should be included on software-type CIs include:

- Software title
- Status of the software item (a list of the different software statuses this could have – design, build, live, retired etc)
- Software provider

Other types of CIs, such as entire systems or individual workstations can have different sets of fields available to complete on them when CIs of these types are created.

## Release and Deployment Management

Release management is very similar to change management in that it has a very clear set of steps that enable the authorization of the release. Releases are triggered by service portfolio activities however rather than normal IT operations or break-fix activity.

The main window used to create a Release contains the following information.

The screenshot shows a web-based form for creating a release. It is divided into two main sections: 'User Details' and 'Release Details'.

**User Details:** This section contains three input fields: 'Raised By:' (a dropdown menu), 'Phone:' (a text input field), and 'Email Address:' (a text input field).

**Release Details:** This section contains several fields:
 

- 'Release Outline:' is a text input field with an orange background.
- 'Release Description:' is a large text area with a vertical scrollbar.
- 'Release Type:' is a dropdown menu.
- 'Proposed Date:' is a date picker showing '05 August 2014'.
- 'Risk Rating:' is a dropdown menu.
- 'Status:' is a dropdown menu currently set to 'Open'.
- 'Release Plan:' is a text input field with a file upload icon (three dots) to its right.

At the bottom of the form, there are three buttons: 'Save As', 'Open', and 'Delete'.

The fields that it's important to include are:

- The person who has requested the release and their contact details
- A summary of what is to be released
- Some more detailed information on what is going to be released
- The type of release (Emergency, Major or Minor)
- The proposed date for the release
- The risk rating of the release (High, Medium, Low or None)
- The current status of the release
- The release plan should be attached as a document or hyperlink

### Actions available on the process

**Submit** – This is the first action available after the release process has been initiated. It confirms that the release plan has been added to the record along with any other pertinent data.

**Add Attachment** – This action is used to add other supporting documentation or files to the record.

**Attachment**

Title:

Attachment Document: **Attachment Document:**

Created: 05 August 2014 18:03   By: SA

Updated: 05 August 2014 18:03   By: SA

**Add Change** – use this action to add the release record to an existing change record.

**Add Note** – Use this action to add any further information to the record that isn't held in a separate document. This may include meeting minutes or summaries of other discussions or related information.

**Note**

Summary:

Details:

Created On: 05 August 2014 18:05   Created By: SA

**Assign/Reassign** – Assigns the release to another group or set of users. This can also be used for notification purposes.

**Assignment**

Group:  Analyst:

Role:  Notify Assignee?  Notify Originator?

Title:

Details:

Report:

Created: 05 August 2014 18:06  By: SA

**Build and Configure Release** – Use this action when the release is ready to be built.

**Build and Configure Release**

Summary:

Details:

Created On: 05 August 2014 18:08  Created By: SA

**Build Failed** – This action is used when a release build fails.

**Build Successful** – Use this action when the build and configuration of a release has been successful. This action is audited and privileged to ensure that only authorized personnel are able to view it.

**Pass for Testing** – This action is used when the release is ready to be tested. It is important to complete the fields on this action to ensure that all the correct details are captured.

**Pass For Testing**

Testing Group:  Testing User:

Test Type:

Test Outline:

Test Details:

Created On: 05 August 2014 18:11  Created By: SA

This action triggers a task process assigned to the specified user or group. The task process itself can ensure that correct test procedures are followed based on the type of testing to be executed. The process cannot move forward until all the task processes are finished.

**Add release activity** – Once all the tasks have passed testing the record moves to a status of 'Testing Successful'. At this status it is possible to add release activities before accepting the release to be rolled out.

**Release Activity**

Group:  User:

Activity Type:  Activity Due Date: 05 August 2014 18:17

Summary:

Details:

Created On: 05 August 2014 18:17  Created By: SA

**Accept Release** – This action requires that the executor completes several fields to confirm that up to this point all due diligence has been performed.

**Accept Release**

Contracts Complete? <input type="checkbox"/>	CIs Available? <input type="checkbox"/>
Funding? <input type="checkbox"/>	Licensing? <input type="checkbox"/>

Notes:

Created On: 05 August 2014 18:19

Created By: SA

**Release Failure** – This action is available throughout the process to be used at any stage.

**Release Failure**

Failure Reason:

Summary:

Details:

Created On: 05 August 2014 18:22

Created By: SA

**Release Roll-out plan** – This action is only available when all associated changes have been completed. It requires that the details of the roll out and the Backout plan are detailed on this record.

The screenshot displays two main sections: "Roll Out Plan" and "Back Out Plan".

**Roll Out Plan Section:**

- Release Outline:** A text field containing "Web Procurement v2".
- Roll Out Date:** A date picker showing "05 August 2014".
- Status:** A dropdown menu set to "Release Accepted".
- Roll Out Plan:** A large, empty text area for detailing the rollout plan.
- Roll Out Plan:** A smaller text field, currently empty.
- Buttons:** "Save As", "Open", and "Delete" buttons are located at the bottom right of the section.

**Back Out Plan Section:**

- Back Out Plan:** A large, empty text area for detailing the backout plan.

**Commit Configuration Items** – This action is used to commit the alterations to all the associated CIs that were affected by the release. This may include the disposal of the CI as well as potential updates to the CI.

**Rollout Failed** – Use this action if the rollout of the release was not successful.

**Rollout Successful** – Use this action when the rollout is successful.

**Close** – Use this action to close off the release record when it has been successfully rolled out. Ensure the closure category is completed (these categories are the same as those for changes).

Closure

Summary:

Category:

Details:

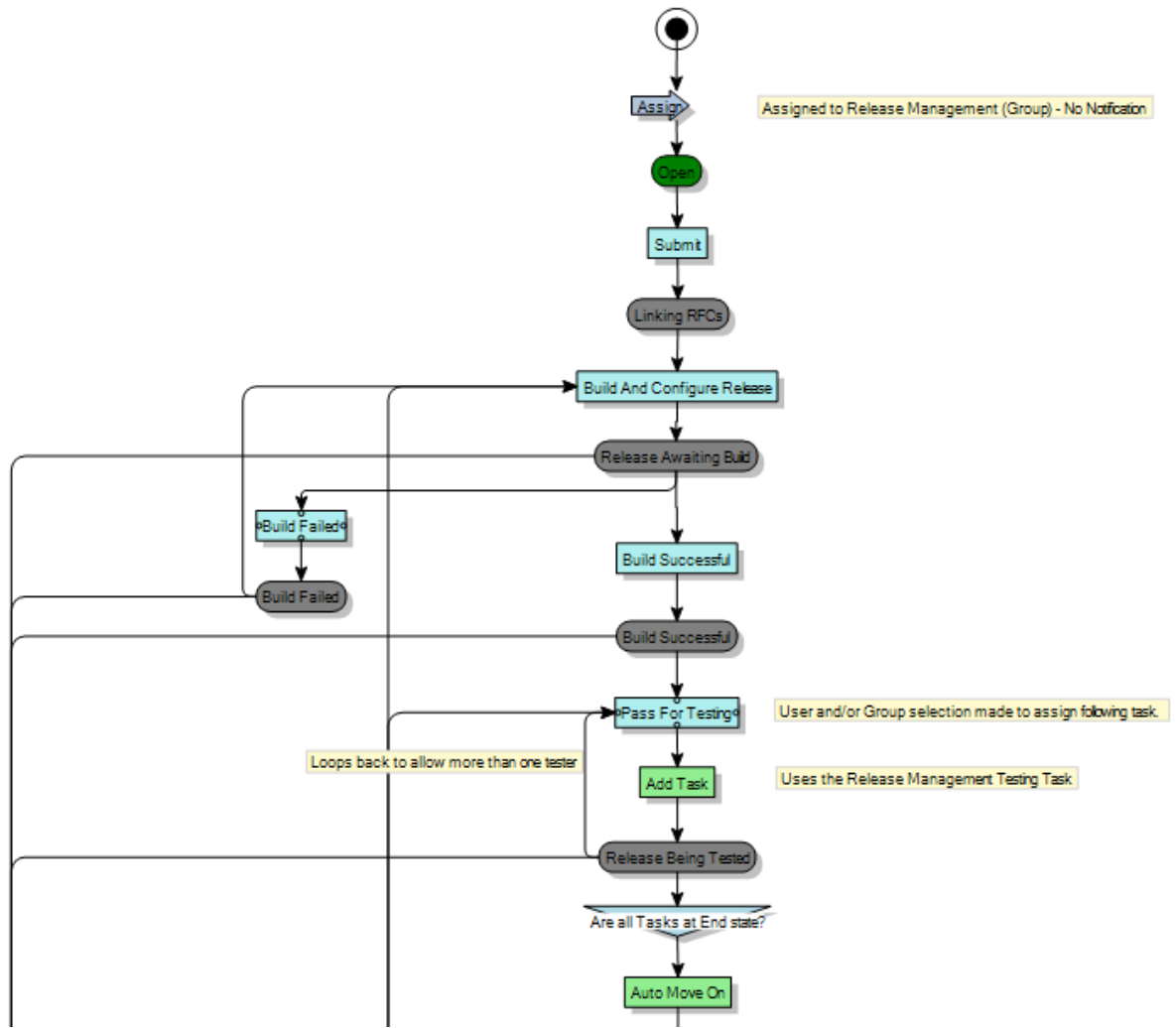
Created On: 05 August 2014 19:14

Created By: SA

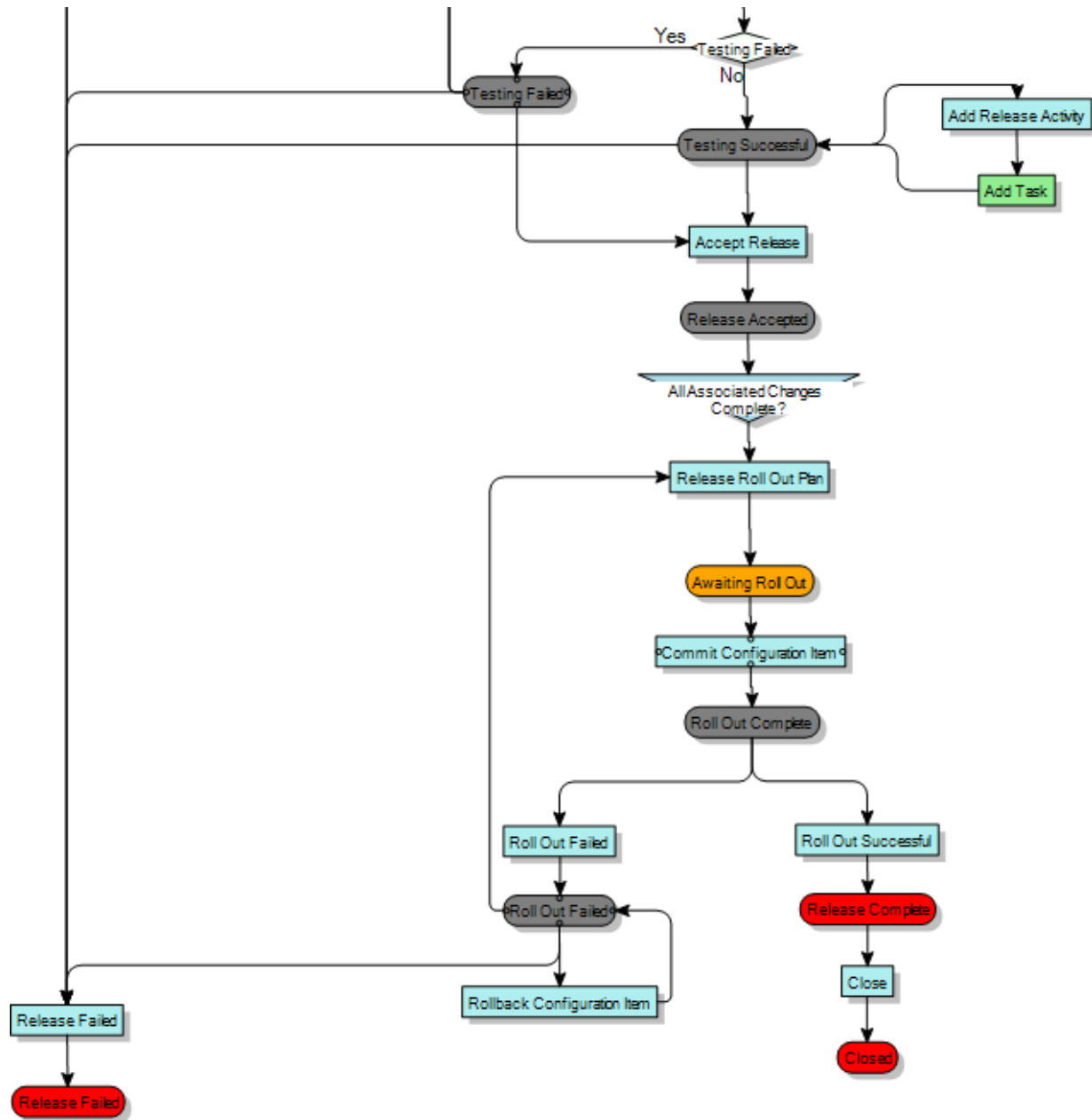
Updated On: 05 August 2014 19:14

Updated By: SA

## Process Design



Continued....



This process is very linear, ensuring that it cannot be moved forward until the relevant documentation has been provided at each stage. However, it can be copied, modified or started again from scratch to support the business requirements of the release processes.

