

Master the problem management process

Keep the business running smoothly with effective problem management

Introduction

This process guide is a detailed explanation of how you can use ServiceNow® Problem Management to manage your problems efficiently. The primary goal of the problem management process is to minimize the impact of problems on the business and prevent recurrence.

Problem management process scope

Problem management includes:

- Identifying and diagnosing problems through event management, technical identification, and proactive problem management
- Diagnosing all problems as quickly as possible using:
 - Problem and error control
 - Event and incident trends
- Identifying known errors and/or workarounds to reduce incident duration
- Identifying and implementing permanent solutions to keep incidents from reoccurring

Problem management process objectives

The objectives of problem management are to:

- Determine the root cause of incidents, identify viable workarounds, and find permanent solutions that prevent recurrence
- Maintain information about problems, known errors, associated workarounds, and permanent solutions
- Communicate information appropriately to reduce and eliminate the number and impact of incidents over time
- Identify problems and solve known errors proactively to improve IT services and prevent potential incidents from occurring

Regardless of your level of maturity with ServiceNow, follow this guide as closely as possible. At ServiceNow, we encourage using the same simple, lean ITSM processes that are reflected in our out-of-the-box designs.

In this guide, you'll find additional recommendations from ServiceNow Professional Services beyond the specific out-of-the-box (OOTB) functionality. You may add additional functionality to what's offered, but you should only do so in scenarios when you will achieve a required

business outcome that can't be achieved using an OOTB method. When you follow this approach, your upgrade paths will be smoother, and you'll be better able to expand your use of the Now Platform®.

This process guide will help you develop and maintain an effective problem management process by explaining:

- The best practices for controlling the lifecycle of problems
- How to streamline the process to support effective problem management
- How to integrate other processes with problem management

BEFORE YOU START, YOU NEED:

- A process owner who is the authoritative voice for the organization and is able to make decisions
- A willingness to critically examine your current working practices

Terms and definitions

Problem – A problem is defined as the underlying cause of one or more incidents. The root cause is not known at the time a problem record is created, and the problem management process is responsible for further investigation. A problem can be:

- The occurrence of the same or similar incidents multiple times
- An incident that impacts one or more services with many users
- The result of diagnostics-revealing systems not operating in the expected way that will lead to an incident

The primary goal of the problem management process is to prevent problems and resulting incidents from occurring, to eliminate recurring incidents, and to minimize the impact of incidents that cannot be prevented.

Known error – A known error is an identified issue that has caused—or has the potential to cause—an incident. You may not know of a workaround or the root cause of a known error. They are not held in a separate known error database—they are identified in the problem record to avoid duplicating records.

Root cause – A root cause is defined as the underlying or original cause of an incident or problem.

Workaround – A workaround is defined as a temporary way to restore service failures to a usable level.

ServiceNow problem management – ServiceNow focuses on using automation and information to speed the path to identifying a root cause and finding a permanent resolution.

Problem management relies heavily on:

- The CMDB for problem assignment and impact analysis
- The incident management process for providing details of individual related incidents
- The change management process for controlling the changes needed to solve problems
- The knowledge management process for sharing information about known errors

Roles and responsibilities

Process owner

The problem management process owner's primary objective is to own and maintain the problem management process. The process owner role is usually filled by a senior manager with the ability and authority to ensure all stakeholders roll out and use the process.

Responsibilities

- Define the overall mission of the process.
- Establish and communicate the process's mission, goals, and objectives to all stakeholders.
- Document and maintain the process and procedures.
- Resolve any cross-functional (departmental) issues.
- Ensure proper staffing and training for execution.
- Ensure consistent execution of the process across the organization.
- Monitor, measure, and report on the effectiveness of the process to senior management.
- Continually improve the process.

ServiceNow Role – There is no role in ServiceNow, because it's a functional role that supports the process.

Problem manager

The problem manager is concerned with the day-to-day activities of the problem lifecycle and is the owner responsible for getting a problem permanently resolved or prevented as soon as possible and ensuring it doesn't reoccur.

Responsibilities

- Manage the day-to-day activities of the process, ensuring operating procedures are documented to support the activities.
- Review problems to check for quality and completeness.
- Assign tasks to the appropriate technical teams for diagnosis and resolution.
- Publish known error articles.
- Provide stakeholder communication on active problems.
- Coordinate decisions on whether to apply a fix.
- Report on problem management process metrics.
- Drive the efficiency and effectiveness of the problem management process.

ServiceNow Role – The **problem_manager** role is required.

Technical support

Technical support teams assist the problem manager in investigating problems and identifying and implementing solutions.

Responsibilities

- Provide subject matter expertise.
- Conduct investigation into problems.
- Identify known errors and the root cause of problems.
- Identify workarounds, and notify service desk and technical support of workaround availability.
- Identify technical solutions to eliminate faults.
- Apply fixes.

ServiceNow Role – The **itil** role is required.

Service owner

Each service must have a designated owner. The service owner is accountable for delivering a specific IT service and is responsible to the customer for the initiation, transition, and ongoing maintenance and support of that service.

Responsibilities

- Create the service delivery method, including engaging and coordinating the teams responsible for fulfillment.
- Ensure the service is being delivered within the expected time frame and cost allocated to it.
- Approve requests against the service.

ServiceNow Role – The **itil** role is required.

Requester

A requester is anyone in the organization making a request. Requesters may submit service requests on behalf of others (recipients).

ServiceNow Role – No role is required in ServiceNow, but requesters need a login.

Approver

Line managers, service owners, and business stakeholders (such as financial management) are responsible for reviewing request details and granting approval or rejecting the request. The actual number and type of approvals required depends on the individual request.

ServiceNow Role – The **approver_user** role is required.

Fulfiller

A fulfiller is an individual assigned to execute a specific task(s) to fulfill the service request.

Responsibilities

- Complete the activities in their assigned tasks.
- Update the task to reflect the current progress and provide detailed information to allow a service desk agent (SDA) to give the requester an update on progress if required.
- Close the task when it's completed with closure information.

ServiceNow Role – The **itil** role is required.

How problems are initiated

Directly in ServiceNow – An SDA, incident manager, problem manager, or other IT user can create a problem directly when they believe they see an underlying issue that's worth investigating.

Directly from an incident – Incident managers, SDAs, problem managers, or other IT support staff can raise a problem from an incident when they're seeing evidence of a problem based on the incident record(s).

Automatically via integrations – Problems can be automatically generated via external systems such as a vendor system integration.

Problem management lifecycle

The states in every ServiceNow application serve a specific purpose. They're designed to make it clear where a particular record currently resides in a process and to display progress. States should represent a unique phase in a process when a specific set of related activities are grouped together and designed to achieve a particular outcome so you can move to the next phase of the process.

Our recommended problem management process has the following state model:

- **New**
- **Assess**
- **Investigate**
- **Remediate**
- **On Hold**
- **Resolved**
- **Closed**
- **Canceled**

Process overview

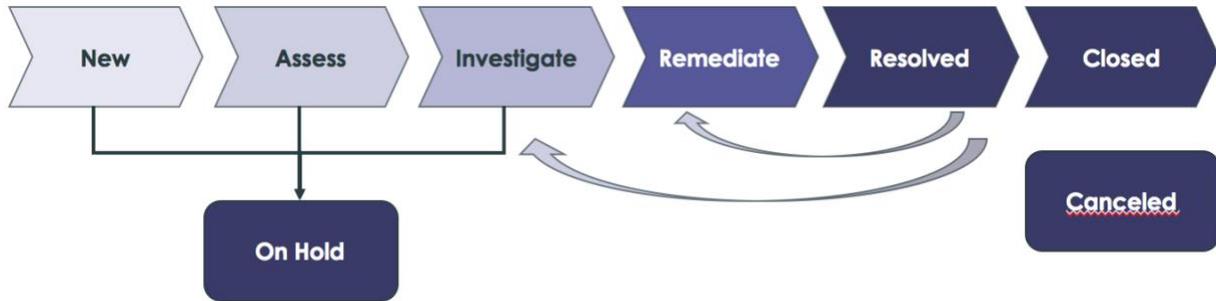


Figure 1: The problem management process overview

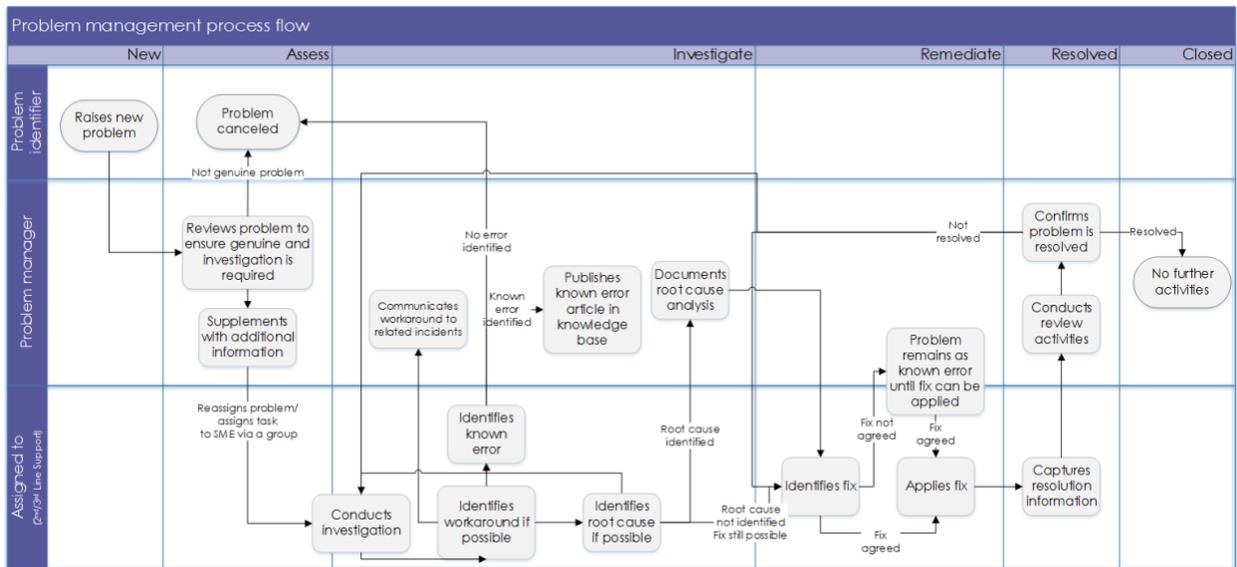


Figure 2: The problem management process flow

State – New

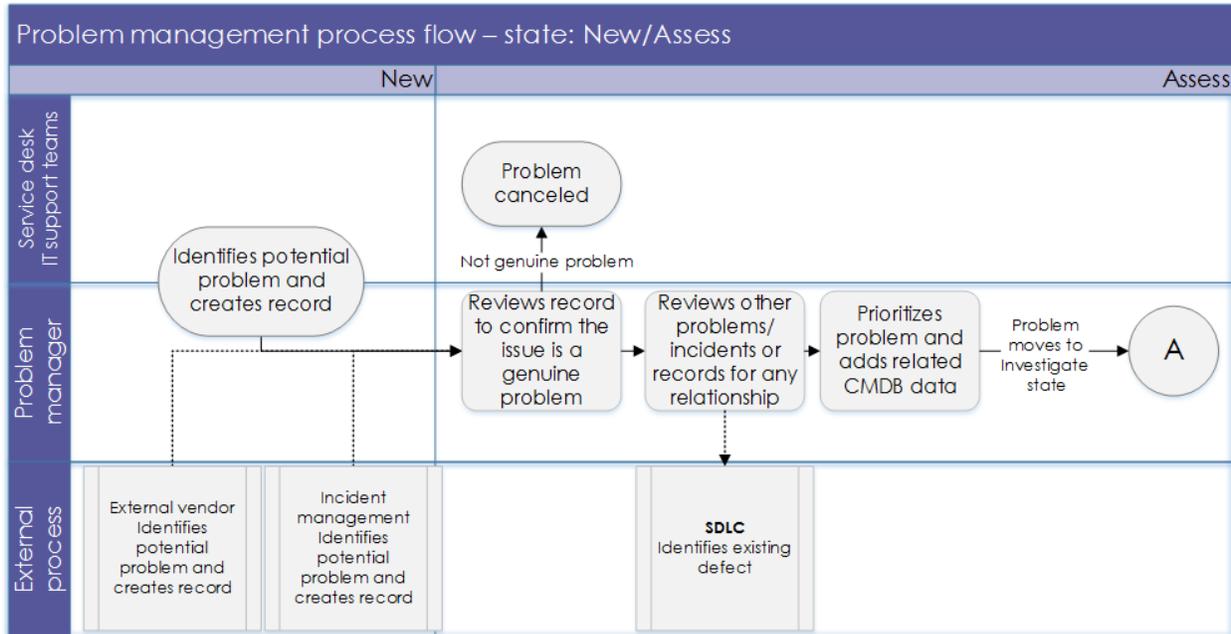


Figure 3: Problem management process flow for the **New** and **Assess** states

When a problem is first created, it's in the **New** state. This is when very basic information is added that may suggest a problem exists. All known information about the symptoms is captured—at the very least, enough information is captured to warrant some kind of investigation.

The mandatory fields are:

- **Assignment group**
- **Short description**
- **Description**

If other fields, such as the **CIs** (configuration items) field are known, they can still be added—and they will be added automatically if they're coming from an existing incident record—but they're not mandatory to move forward.

Problem assignment

In the **New** state, you must identify the appropriate assignment group to assess the problem in the next phase of the lifecycle. This needs to be a problem management group or, if your organization doesn't have a dedicated problem management function, then the group that will perform that function—possibly the service owner or incident management team—requires the **problem_manager** role.

Expert tip

This assignment is best achieved by automatically updating the **Assignment group** field rather than letting the user try to pick the correct group manually since this approach is prone to error.

Once the mandatory fields are populated, change the **State** field to **Assess** or click the **Assess** button. This will move the problem into the lifecycle where it is considered “live” and something that requires attention.

State – Assess

In the **Assess** state, the problem manager is primarily assessing the problem to determine whether it's a genuine problem or not.

Here's the process:

1. The assignment group (your problem management group) chooses an individual to take responsibility for the problem on behalf of the group. To do this, they populate the **Assigned to** field.
2. Next, the individual assigned the problem (the problem manager) conducts an initial review of the problem primarily to ensure that it is, indeed, a genuine problem.
 - a. If the problem manager recognizes that it's not a genuine problem, they change the **State** field to **Canceled** and provide a **Work note** entry to explain why they do not consider the problem to be genuine and something that requires further investigation.
 - b. If the problem manager recognizes the problem is genuine, they populate the fields that are mandatory in the **Assess** state.

Mandatory fields for the **Assess** state:

- **Impact** – The affect that a problem has on the business
- **Urgency** – The extent to which the problem's resolution can bear delay
- **Priority** – Automatically populated from the **Impact** and **Urgency** fields according to Table 1
- **Business Service OR CI**

Establishing priority

Problem prioritization typically drives the criticality associated with handling a problem and the order you'll follow to handle problems. This can be accomplished in two ways.

The problem's priority can be calculated according to its impact and urgency.

		Urgency		
		1 – High	2 – Medium	3 – Low
Impact	1 – High	Priority 1 – Critical	Priority 2 – High	Priority 3 – Medium
	2 – Medium	Priority 2 – High	Priority 3 – Medium	Priority 4 – Low
	3 – Low	Priority 3 – Medium	Priority 4 – Low	Priority 4 – Low

Table 1: How a problem's impact and urgency define its priority

Or the problem's priority can be automatically established based on the CI identified in the problem record. With this technique, the business criticality value of the CI is used to determine the problem's priority.

Here's an example of establishing priority based on its CI: To a financial organization, its online banking service is considered critical. If the online banking CI is related to the problem, the system can automatically set the priority to **Critical** as a result.

Expert tip

When you automatically establish priority based on a problem's CI, you can more accurately and consistently prioritize problems since determining impact and urgency can be subjective. If you use this automated method, it can occur in the **New** state when the problem is first raised—knowing its urgency immediately is helpful to the problem manager.

Business services and CIs

The **Business service** and **CI** fields are used to identify what the problem is impacting. These fields use CMDB data, which is also used across other ITSM processes, creating a valuable link and traceability, particularly for business services.

Completing these two fields allows you to use the Dependency Views feature. With this feature, you can see a map of the relationship between the problem's service or CI and other related CMDB components. It also shows any ongoing incidents or changes that may exist, with root cause analysis.

Once the problem manager enters this data, they change its state to **Investigate**.

During the **Investigation** state, three main activities should occur:

- The problem is identified as a known error (something that has or potentially will cause an incident), or it will be discounted completely. If a known error is not found, the problem is canceled with a **Work note** entry to explain why.
- If the problem manager or assigned team discovers a known error, they select the **Known error** check box. They can also communicate the known error to the service desk or end users by creating a known error article in the knowledge base. To do this, they click the **Post Knowledge** related link. This creates a new knowledge article in the **Draft** state and will automatically include the **Description** and **Workaround** field information.

Note: The **Description** and **Workaround** fields can be removed or edited to suit the audience, particularly if the information is very technical.

- If it is not deemed necessary to publish the known error in the knowledge base, SDAs and other technical support users can search for all problems with the **Known error** check box set to **True** to find them. Most commonly, SDAs search for known errors when they're trying to resolve an incident, find a workaround, or if they recognize a pattern of similar incidents.

Set the **Known error state** field. If investigation is still ongoing, select **Root Cause Under Investigation**.

You may also discover a workaround during the **Investigation** state. If so, enter it in the **Workaround** field. You can communicate the workaround to all open related incidents by clicking the **Communicate Workaround** related link. This populates the text from the **Workaround** field into the **Activity Log** of all related open incidents, explaining that it's a workaround from the problem record.

Finally, discover the problem's root cause and document it using the **Root cause code** and **Root cause detail** fields.

All these activities can happen in parallel or may be discovered at different times through investigation. You may discover none, some, or all three.

If you discovered a known error and performed a lengthy investigation but still can't identify the root cause or find a workaround, the problem manager may end the investigation. When that happens, set the **Known error state** field to **Investigation Suspended**. The problem will remain in this state indefinitely so it remains documented and searchable. In the **Investigation Suspended** state, users can continue to relate incidents to it.

If you discover the root cause or a workaround, click the **Remediate** button to move the problem to the **Remediate** state.

State – Remediate

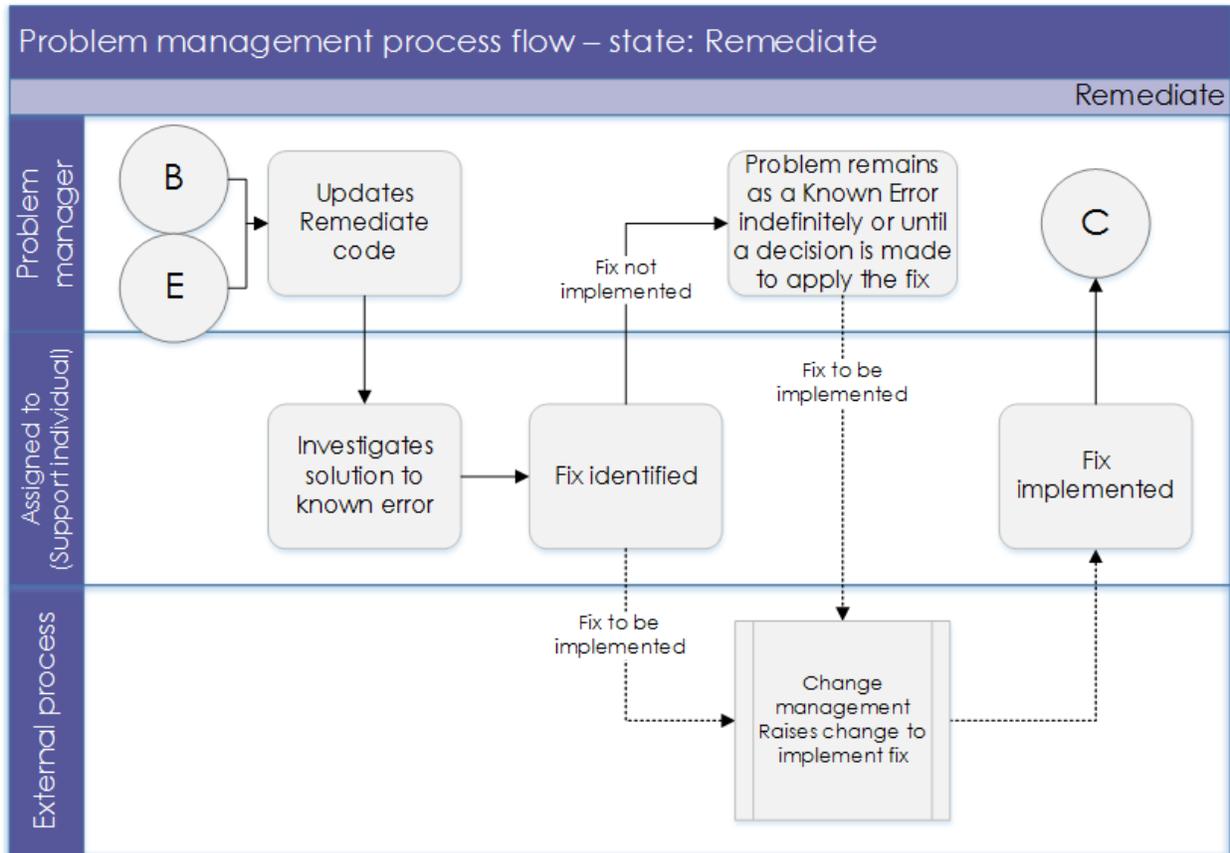


Figure 5: Problem management process flow for the **Remediate** state

Problems in the **Remediate** state have been investigated and now need a fix. In this state, you must enter a new **Known error state** with one of the following choices:

- **Fix Being Identified**
- **Fix In Progress – Awaiting Change**
- **Fix Deferred – Cost**
- **No Fix Available**

Initially, the **Known error state** field is most likely set to **Fix Being Identified** since the next activity is to try and understand how to fix the known error.

To identify the fix, either:

- Keep the problem record assigned to the correct group and individual to identify the solution
- The problem manager assigns individual task/s, as you can do during the **Investigate** state.

Use **Work note** entries to log the actions that have taken place and, once it's identified, information about the fix.

When the problem's fix is identified, the person assigned to the problem must make a decision about whether or not to implement the fix and changes the **Known error state** to one of these states:

- **Fix in Progress – Awaiting Change** – Select this state when the fix is planned to be applied. This triggers a requirement to relate a change request to the problem to show its clear path to resolution.
- **Fix Deferred – Cost** – The person assigned the problem may also decide not to fix the known error due to cost implications. Many organizations will recognize that the cost of fixing it is simply not worth the benefit and are content to live with the known error. If this is the case, select this state.
- **No Fix Available** – After a full investigation, there may be no identifiable solution. When this occurs, select this state. When the known error won't or can't be fixed, the problem process ends here. The problem record remains open and marked as a known error with a **Known Error state** clearly identifying that it is not being fixed. This record remains easily searchable so other users can relate any future incidents that are raised as a result of the known error to the problem.

The problem manager can review the known errors that won't be fixed at appropriate intervals to establish whether the impact has increased or the situation has changed to such a degree that either:

- The known error will be fixed. In this case, update the **Known error state** field to **Fix in Progress** and apply the change following the standard process.
- It's no longer a known error due to some other change that has occurred and resolved it.

Once the change is implemented and the problem is considered resolved, click the **Resolved** button to move the problem to the **Resolved** state.

State – On Hold

Use the **On Hold** state to indicate that a problem is not yet resolved but is temporarily not being worked while it waits for further action to occur outside of the control of the individual assigned to it. This doesn't include where a change is being implemented to fix the known error.

There is a separate **On hold reason** field that shows why the incident is on hold. This is a free-text field so users can explain why they're putting the problem on hold.

To move the problem to the **On Hold** state, either change the **State** field or click the **On Hold** button. This makes the **On hold reason** field mandatory.

If your organization uses SLAs or OLAs in problem management, this state typically acts as a pause condition for them. Once someone changes the state from **On Hold**, the **On hold reason** field is cleared, since it is no longer on hold and it's possible to put the problem back on hold for a different reason later. All **On hold reason** entries are captured in the **Activity Log**.

State – Resolved

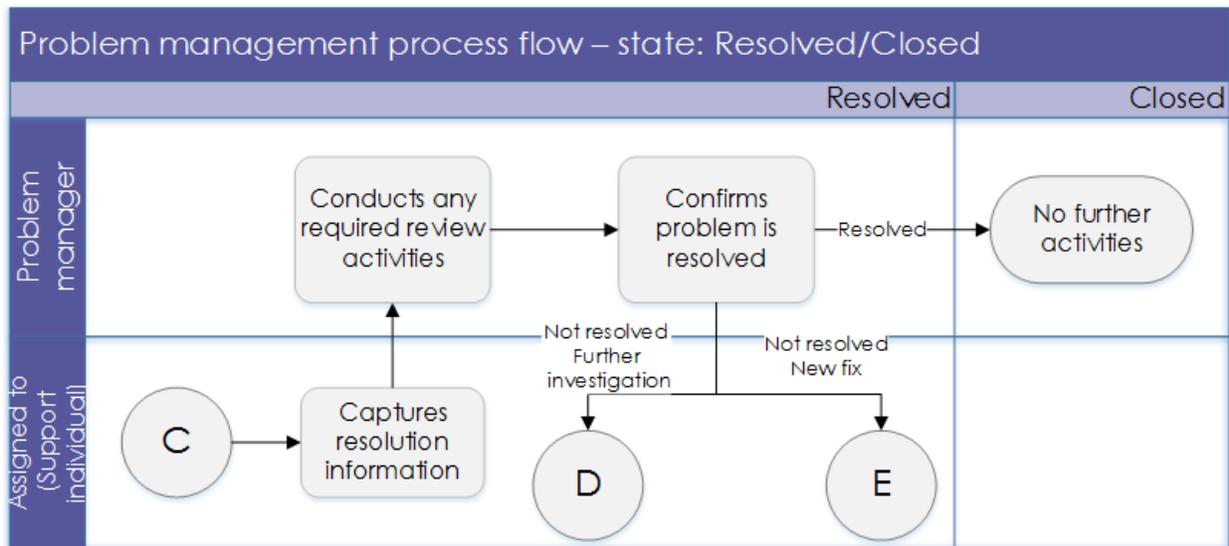


Figure 6: Problem management process flow for the **Resolved** and **Closed** states

Once the problem has been moved to the **Resolved** state, the individual assigned to it must populate the mandatory **Resolution code** field with one of these two choices:

- **Resolved – Fix applied**
- **Resolved – Workaround applied**

Next, populate the mandatory **Resolution notes** field with a description of exactly what was done to solve the issue.

In the **Resolved** state, an organization can conduct a review of the problem if its process requires it, and you can add additional fields to capture that information. Alternatively, the problem manager can assign problem tasks as required.

Problem managers or others assigned to the problem can observe a set period of time before they set the problem to the **Closed** state to confirm that the known error has definitely been solved. If evidence suggests that the issue persists, they can set the state back to **Investigate** or **Remediate**, whichever is the most appropriate for the situation. Then, they can work through the process again to continue looking for the solution.

If they can confidently consider the problem solved, the person assigned to the problem will click the **Close** button to change the state to **Closed**.

State – Closed

No activities take place at this state. Should the problem reoccur, a new ticket must be raised. Once a problem is closed, it cannot be reopened.

State – Canceled

There are very few scenarios when a problem is genuinely canceled. This only occurs when:

- Someone raised a problem in error—likely before realizing there is no real problem

- The problem is a duplicate of an existing problem that was identified later

When you cancel a problem, you must enter an explanation for the cancellation in the **Work note** field. Canceled problems cannot be reopened.

Other processes

Incident management

The majority of problem records are triggered in reaction to one or more incidents. Use the incident history to help you identify trends or potential weaknesses as part of proactive problem management.

Incident records that are related to problem records that are pending resolution are automatically updated when a problem is resolved.

Change management

For problems, implementing the workaround or the permanent solution requires work on a service, hardware, or software. Conducting this work requires raising a change record. To do this, select the **Create Normal Change** or **Create Emergency Change** option from the drop-down menu.

Emergency changes typically require a related incident record to prove that they're urgent enough to bypass the full process and lead times.

Configuration management

The configuration management system underpins all records and activities related to any CI. It contains details of the infrastructure vital to services, CIs, and their relationships.

The CMDB is used within the problem management process by relating CIs, including business services, to the problem. This allows you to use the Dependency Views feature to display the relationship between the selected CI and other CIs related up and downstream.

Asset management

Information about attributes of assets and configuration items may be used in various Service Catalog items that initiate service requests. Updates to certain asset and configuration item attributes (for example, software license counts) may be triggered from a service request.

Knowledge management

Knowledge is a vital part of the problem process. Known errors are documented and published in the knowledge base, along with their workaround information, to allow service desk agents to find information and assist with other incidents related to the same known error. End users can also see knowledge articles and, if appropriate, make use of the workaround information while the known error is being fixed.

Process governance

Measurement

Key performance indicators (KPIs) evaluate the success of a particular activity toward meeting its critical success factors. You can successfully manage KPIs either by repeatedly meeting an objective (maintaining the KPI) or by making progress toward an objective (increasing or decreasing the KPI).

The Benchmarks feature in ServiceNow gives you instant visibility into your KPIs and trends, as well as comparative insight relative to your peers' industry averages. You can contrast the performance of your organization with recognized industry standards, and view a side-by-side comparison of your performance with global benchmarks.

Heads up!

When you create any KPIs or metrics for problem management, don't be driven by the same measurements you used for incident management. With incidents, the purpose of the process is to restore service as quickly as possible using whatever means available—speed of resolution is a key measurement for this process. With problem management, the purpose is to understand the underlying cause of issues and permanently fix them no matter how long that takes—if it's even possible.

You shouldn't measure speed of resolution in problem management. It would drive the wrong behavior for the process and focus on closing records rather than finding a permanent fix. Process owners need to feel comfortable with problem records potentially remaining open for months or even years.

Metrics

Process KPIs

Process KPIs provide information on the effectiveness of the process and the impact of continuous improvement efforts, and they are:

- Best represented as trend lines and tracked over time
- Monitored by the process owner

Item	Purpose
Mean time to first respond to problems, by priority	This KPI measures of how well response SLAs are achieved.
% of problems with a root cause identified for the failure	This KPI measures the effectiveness of problem management in defining a root cause.
% of problem with workaround defined	This KPI measures the effectiveness of problem management in defining and communicating workarounds.
Percentage of incidents resolved by fixing known errors	This KPI measures the effectiveness of problem management in supporting the timely resolution of incidents.

Table 2: Some process KPIs and their purpose

Operational data

Active catalog items or requests that require visibility, oversight, and possible management intervention are best tracked on a dashboard or homepage that is monitored by the service desk and request fulfillment team.

Item	Purpose
Problems ready to be assessed	This shows all problems that require a problem manager to assess them.
List of active problems that have missed target response times	This highlights where there may be a process issue in assessing new problems.
Aged list of backlogged problems	This gives you visibility to unassigned work.

Table 3: Some operational KPIs and their purpose

The takeaway

Keep these things in mind as you continue to improve your problem management process:

- Don't use problem management as a parking lot for incidents that can't be solved.
- SLAs are normally not associated with problems since they never close.
- Problem investigation and root cause analysis can deliver quantifiable cost savings to the business.