



ITIL Change Management - A Beginner's Guide

Understand the basics of Change Management



www.freshservice.com

Table of content

ITIL Change Management	1
Change Management Process flow	3
Types of Changes	6
Use cases for ITIL Change Management	8
Integration with other modules	11
The Right Approach	13

ITIL Change Management

Digital transformation and initiatives lead to new projects within organizations. Seamless business operation is a gift for businesses to improve productivity. Businesses undergo transitions on a regular basis that differ in risk and impact. It is vital to streamline this process to keep these two factors in control. Change management is significant to deploy new changes without any disruption or downtime. ITIL change management follows a standard operating procedure to eliminate any unintended interruptions and capture necessary details about a change before it is implemented such as reason for change, planning and approval.

ITIL Change management is essential for businesses to implement changes smoothly and maintain current working state. Change management works closely with other ITIL modules such as Incident management, problem management, configuration management to manage infrastructure and Configuration Items, CIs that are affected or going through the change. This delivers better context and consistency so that change team can be proactive in avoiding any potential failure. Implementing a change is a costly affair thus change management ensures everything is in control and even if something goes wrong, damage is minimal.

Objectives

- Mitigate risk and impact
- Retention of current working state
- Communication and approval management
- Effective change planning using available resources
- Reduction in number of incidents due to change



Shifting office space, deploying a fix to production server, windows patch, replacing cloud service provider, updating OS.

Change Management Process flow

ITIL Change management follows a set of processes and every detail about change is recorded for future tracking. Following the process ensures that there are no loopholes and change is validated to ensure successful deployment. It is helpful for other service desk teams such as release management, configuration management to understand the type and complexity of change. Change manager is responsible for successful planning & evaluation. Release management team takes care of the actual implementation of change. Following is the change management process flow

Request for Change

The first step is to request for a change with valid reasons. Change requests are created due to one of the following reasons

- Incident causes a new change
- Change is created as a result of a known problem
- End user requests a new change
- Change manager creates a change as result of an ongoing maintenance

Request for Change proposal contains following information along with necessary details



Reason for change

lists the reason on why change has to be created, affected parties and desired outcome.



Impact & Risk assessment

Impact and risk are calculated and documented including configuration items, CIs.



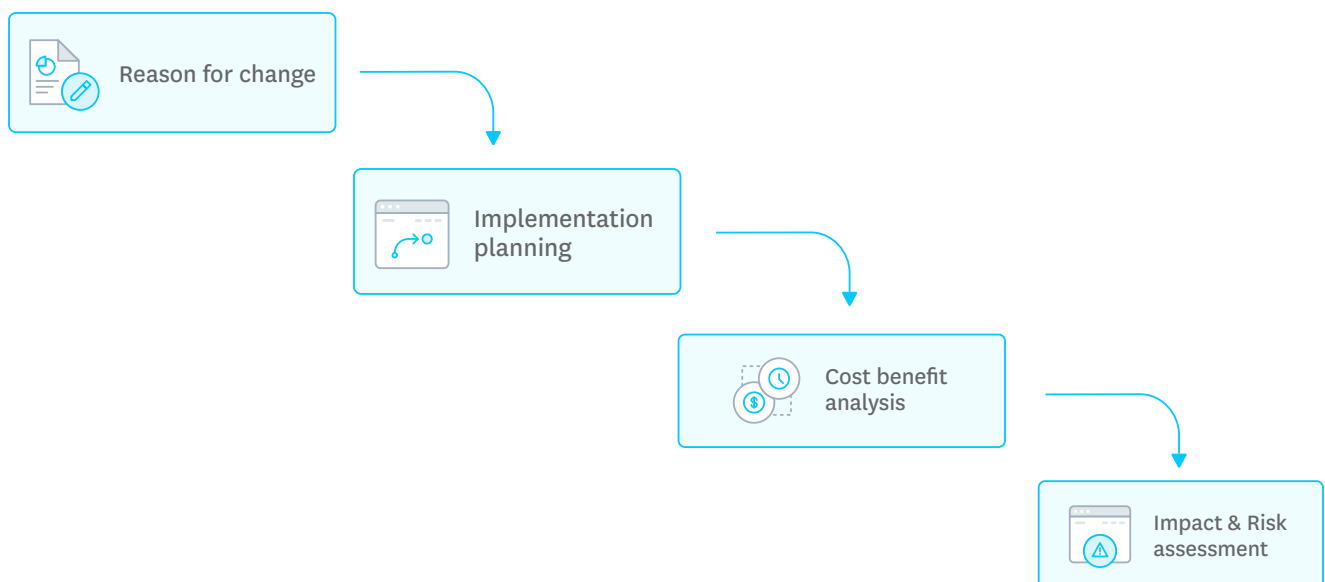
Cost benefit analysis

Estimated resource utilization and cost incurred along with potential benefits are documented



Implementation planning

Steps for implementing change that includes project members, timelines and methodology.



Change Evaluation and planning

Change assessment committee evaluates submitted RFC and suggests necessary changes that would be taken care by Change initiator. This is followed by Change planning that follows a standard procedure and it includes activities such as

- **Prioritizing a change** After analyzing the reasons and RFC, prioritize change request and determine change type depending on the risk/impact
- **Scheduling a change** Depending on the priority, scheduling is fixed. If it's a low priority change, it could be pushed to the next deployment window.
 - Decide on the planned start date and end date
 - Ensure there is no clash with other major activities that are scheduled
- **Roll out plan** details about implementation steps and approach
- **Identifying stakeholders** Identifies the project members who are responsible for carrying out the implementation and members who need to approve

Change approvals

Communicating the change and its details to relevant stakeholders and getting on time approval is key to success. Automate the approval process to reduce manual effort. Approval process flow is decided depending on the type of change. For example, major change requires approval from CAB as well as management whereas standard change does not require any CAB approval as they are pre-approved. Change request is approved only if all the CAB members approve it. Upon rejection, reassessment review is done and submitted again for CAB approval.

Change implementation & review

Once the change is approved, implementation is carried out with the help of release management team. Release team follows their own processes that include planning and testing. Change review happens once implementation is completed to determine whether it's a success or failure. Review of completed changes help in revisiting and modifying existing change management process if necessary.

Types of Changes

Types of changes

Major

Major changes are high impact and high risk items that may alter production systems. This requires CAB approval along with business approval. This has a huge impact on ongoing business operations and also has financial implications. Therefore, RFC contains a detailed proposal on cost-benefit, risk-impact analysis.

Examples - migration from one datacenter to another; replacing an existing enterprise solution(ERP).

Standard change

Standard changes are generally pre-approved changes that have low impact and low risk. These changes occur periodically and follow a standard procedure. They do not follow the conventional process flow and it can be saved as a standard change template for reuse. Every time, CAB approval is not required as these changes are evaluated and approved once initially.

Examples - OS upgrade, deploying patch, setting up an user account etc.

Minor

Minor changes are generally normal changes that do not have a major impact and are less risky to execute. These are non-trivial changes that do not happen frequently but this undergoes every stage of change lifecycle including CAB approval. It is important to document related information so that this can be converted to a standard change in future.

Examples - application performance improvement, website changes.

Emergency

Emergency changes are unexpected interruptions that need to be fixed as quick as possible. This does not follow the usual norm whereas retrospective RFC has to be submitted post implementation and approvals can be managed through Emergency CAB (ECAB). These changes can be reviewed later to avoid potential infrastructure risks in future and detailed documentation is done post change execution.

Examples - Fix for security breach, server outage.

Use cases for ITIL Change Management

Use cases for ITIL change management

There are many practical use cases for ITIL change management that are handled by IT and DevOps team on a day to day basis. Let us look at some of the common use cases.

CASE I - Business Continuity Plan & Disaster Recovery

Application of ITIL change management in BCP/DR is vital as this is one of the critical operations to ensure business continuity. This is classified as a major change due to its high impact/high risk on live environments. This involves changes to production servers and therefore having an effective change management is necessary. The objectives of change management in BCP/DR include

- Minimal/no disruption to the current ecosystem
- Effective resource utilization
- Having an approved backout plan and related backout activities
- Identifying configuration items that may be potentially impacted due to BCP/DR
- Proper documentation of all relevant changes
- Communication of changes to all stakeholders
- Monitoring service availability & metrics such as recovery time taken (RTO,RPO)

BCP/DR has a predefined template to capture necessary details.

Stage	BCP/DR
Request for change	Customer reports an issue and requests for a change that may be due to BCP/DR change implemented. Change usually includes infrastructure changes.
Change planning & evaluation	Change manager & team reviews the RFC; analyzes reason, impact, cost associated and plan for the change in a scheduled time frame. BCP/DR could be applicable to multiple locations. Change team should identify specific location where BCP/DR will be executed.
Change approvals	Get the approvals from BCP team and other stakeholders; Communicate the new changes to all customers, vendors; update all relevant collaterals including BCP template
Change implementation & review	Implementation is carried out by release team and change team performs periodic audit.

CASE II - Security patch

Fixing any security threat or breach is a typical example of an emergency change. This is generally high risk, high impact change with time constraints. Live environments and production systems are vulnerable when you execute emergency changes. Though the fix has to be really quick, standard testing and approval by ECAB is important to avoid any unintended issues. Since emergency changes need quick resolutions, this cannot wait too long to follow a typical change management process. Following is the usual process for emergency changes

- Change requester submits an emergency change ticket
- Change team evaluates the risk/impact
- Quick change planning is done using the template including roll out and back out plan activities
- Approval is triggered to ECAB members (Emergency CAB members)
- Once it is approved, change is implemented
- Detailed assessment happens post change implementation
- Retrospective RFC is submitted post implementation for documentation and audit purpose
- Periodic audit review is done to avoid any similar issue in future

CASE III - Cloud change management

Businesses are moving towards cloud and agile. Change management in cloud is a typical example of standard change which follows a pre-approved procedure. Release cycles have shortened leading to frequent feature releases and enhancements. Deployment in cloud environment often requires smaller units of testing and due to this, changes to production environment might happen often. Therefore, following a standard change management process helps in controlling the risk and impact.

- Create a change request in order to make any changes to production infra.
- If there are any corresponding tickets or problems, associate them to this change ticket.
- Understand the contract with cloud service provider regarding uptime, service availability and security
- Prepare a standard change template for new releases
- Plan your releases - In a multi-tenant architecture, releases need to be planned without any disturbance to existing configuration settings
- Plan for backout activities - To retain current ecosystem, plan for backout activities and keep them ready
- Use sandbox environment for testing - Sandbox testing ensures expected and actual behavior of new releases match with each other without affecting live environment. It reduces risk of changes to production systems.
- Periodic review ensures there is no unexpected interruption during standard change execution

Integration with other modules

Integrations with other modules

Change management is seamlessly integrated with other ITIL modules such as incident management, problem management, release management and CMDB. Information flows across all these modules to maintain consistency.



Change management & CMDB

Configuration Management Database, CMDB contains information about assets and their configuration items. Change implementation might have an impact on configuration items. Therefore, it is essential to associate the right CIs that might be impacted due to change or caused the change to occur and update this information within CMDB. CMDB maintains asset relationships and this information helps in impact analysis of a change. If a particular asset has multiple dependencies or relationships with other assets, then the impact is high and respective change process is defined based on this analysis.



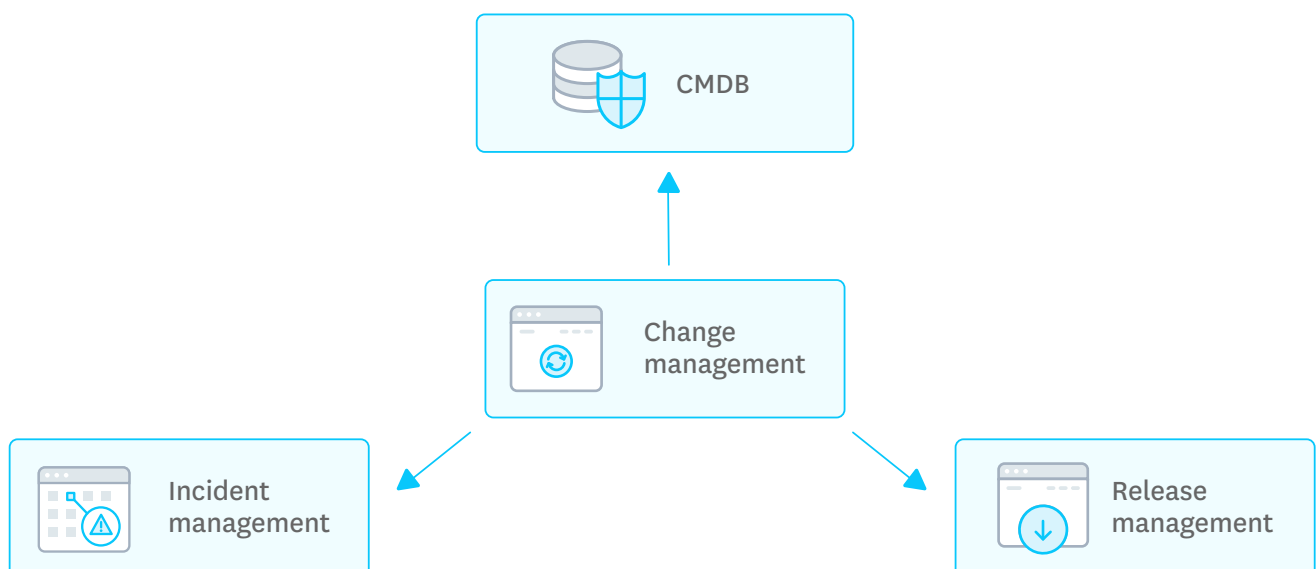
Change & release management

Change management ensures that planned change is under control whereas release management takes care of the actual implementation of planned changes. It is vital to clearly differentiate the roles & responsibilities of change and release management. Successful change leads to new release and change management coordinates with release for build, test and implementation plan. Change review happens post deployment of releases.



Change & Incident management

Major incident may give rise to a new change or incidents may be created due to change execution. If root cause of the incident is already known, then it is linked to a change record. If the cause is unknown and not apparent, then incident is linked to a problem to perform root cause analysis. Problem management finds out a workaround if the issue cannot be solved permanently and this is followed by change management. The goal of incident management is to restore services quickly and change management helps in fixing the issue permanently by rolling out new changes.



The Right Approach

Change management Implementation Checklist

- ✔ Define change templates to pre-fill standard changes thus saving time
- ✔ Define workflows to execute set processes
- ✔ Include stakeholders from all relevant teams in CAB committee
- ✔ Communicate clear roles and responsibilities of every CAB member
- ✔ Automate CAB approval process to eliminate manual work
- ✔ Define success factors and metrics to measure
- ✔ Integrate with other ITIL modules for consistent information flow

The right Approach

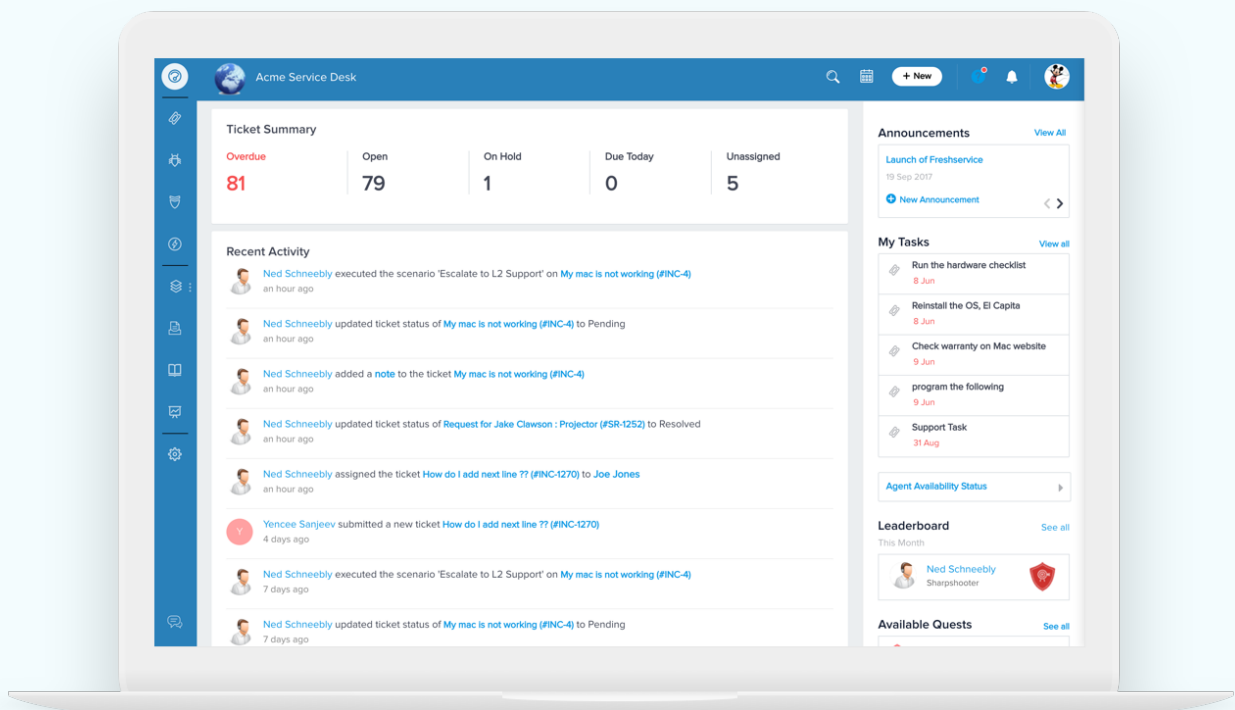
There is no one-size-fits-all approach for change management process. Businesses need to understand their structure, objective, complexity and define change process flows accordingly. However, change management is essential irrespective of the size of business in order to minimize disruptions while undergoing a change and track every change request with proper details.

About Freshservice

Freshservice is a cloud-based IT service desk and IT service management (ITSM) solution that is quick to setup and easy to use and manage. Freshservice leverages ITIL best practices to enable IT organizations to focus on what's most important - exceptional service delivery and customers satisfaction. With its powerfully simple UI, Freshservice can be easily configured to support your unique business requirements and integrated with other critical business and IT systems. Are you trying to keep up with the current ITSM trends? Freshservice is on a constant mission to innovate and deliver great experience.

Start your free trial now :

www.freshservice.com/signup



Visit www.freshservice.com for the latest in the world of ITSM.

Author: Padmavathy Sankaran

Copyright © Freshworks Inc. All Rights Reserved