Insert logo here

(transparent PNG recommended)

SALESFORCE SALES CLOUD  
Release Management Policy

**A NOTE FROM CLOUDKETTLE:**

A Salesforce release management strategy increases productivity and deployment velocity while decreasing costs and downtimes due to changes in the system. Often organizations struggle to produce a high throughput of functionality in Salesforce with a low risk of disruption.

A Salesforce Release Management Policy is necessary to support a well-defined release management process that allows a company to streamline changes from development to production. This template was built for small and mid-sized enterprises and should be populated with your organization’s relevant information in the highlighted areas.

**Once completed, remember to update the Table of Contents and delete this instruction box from the cover page.**

Salesforce Sales Cloud Release Management Policy

**Organization:** **insert company name**  
**Org ID: insert OrgID**

**Policy:** Salesforce Sales Cloud Release Management Policy

**Effective Date:** **insert date**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Revision** | **Rev. Date** | **Description** | **Prepared By** | **Approved By** | **Approved Date** |
| **1** |  |  |  |  |  |
| **2** |  |  |  |  |  |

##### 

Contents

[Purpose 4](#_Toc75427848)

[Scope 4](#_Toc75427849)

[Application to Parties 4](#_Toc75427850)

[Administration 4](#_Toc75427851)

[Release Management Overview 5](#_Toc75427852)

[Requirements Gathering and Planning Process 5](#_Toc75427853)

[Triaging Salesforce Requests 5](#_Toc75427854)

[Business Requirements Document and Product Backlog 5](#_Toc75427855)

[Sprint Planning and Scope 6](#_Toc75427856)

[Business Requirements Document Template 7](#_Toc75427857)

[Salesforce Sandboxes 7](#_Toc75427858)

[Release & Meeting Cadence 9](#_Toc75427859)

[User Acceptance Testing (UAT) 10](#_Toc75427860)

[Communication Plan 10](#_Toc75427861)

[Release Notes 10](#_Toc75427862)

[Updates 10](#_Toc75427863)

[Exception Requests 10](#_Toc75427864)

[Approved Exceptions 11](#_Toc75427865)

# Purpose

This policy exists to set out the principles for coordinating releases in insert company’s Sales Cloud org insert OrgID instance.

This policy aims to mandate guidelines for how changes are pushed from development to production to decrease disruption to the team while maintaining a high throughput of functionality.

# Scope

The policy covers the production environment insert OrgID, along with any sandboxes linked to this production org.

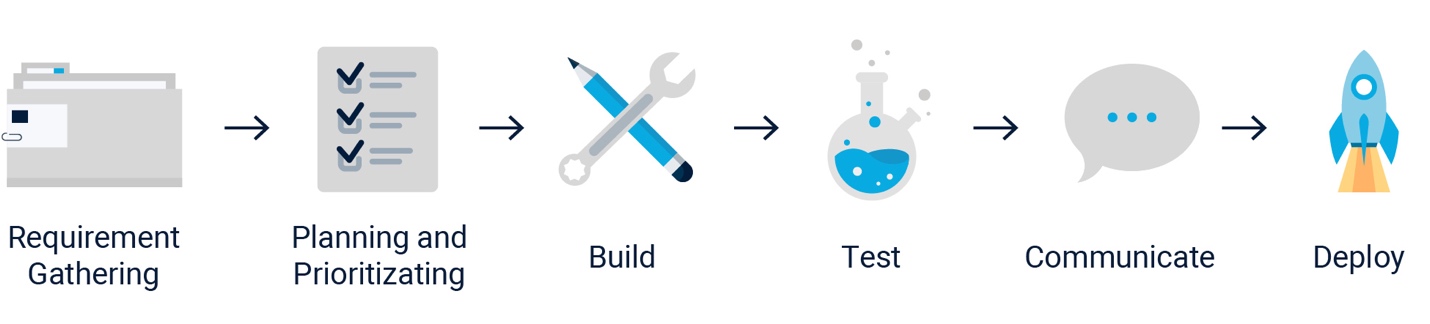
## Application to Parties

This policy applies to all insert company officers, employees, partners, contractors, and services providers that may have access to the Salesforce org. It is the responsibility of all the above parties to familiarize themselves with this Policy and ensure compliance.

# Administration

This policy is administered by the insert company insert job title and managed on a day-to-day basis by the insert name of who will manage this on a day to day basis.

# Release Management Overview



The release management process covers the following areas in the development lifecycle:

* Requirements Gathering
* Planning and Prioritizing
* Build
* Test
* Communicating to end-users
* Deployment

# Requirements Gathering and Planning Process

## Triaging Salesforce Requests

For more information on this topic see Salesforce’s documentation at: <https://sforce.co/37Ayn96>

Salesforce is a central system for many users, so often Salesforce team members get overwhelmed with requests for changes and enhancements. To ensure the team can better prioritize and deliver at a high velocity, a formal requirements gathering and planning process is needed.

## Business Requirements Document and Product Backlog

Users should leverage a business requirement document to request additional functionality in Salesforce. The document requires:

* Details of the request
* The business impact of their request (including justification of importance)

In addition, business impact details and any current workarounds will need to be documented to help the Salesforce product owner prioritize all the requirements.

This document should be readily available to assist users, and the email account salesforce@xxxxx.com should be set up to receive all requests.

These requests will make up the ‘Salesforce Product Backlog’ and will be ‘groomed’ before entering any sprint and being worked on by the Salesforce team.

An issue and product management tool such as **Jira** is recommended to organize the Salesforce Product Backlog.

The Salesforce product owner’s responsibility is to prioritize the items in the ‘Salesforce Product Backlog’ by working with the various stakeholders and ensuring that the requirements are well documented. Each item in the backlog must have an effort and priority assigned to it.

## Sprint Planning and Scope

Before each sprint, there will be a sprint planning session where priority items in the ‘Salesforce Product Backlog’ are properly groomed and added to the sprint backlog.

This practice can be time-consuming at first but is critical to understanding departmental needs and planning for future success. Operating within a ‘Salesforce Product Backlog’ and sprints will also allow all incoming requests to be prioritized and visible across the organization. For more information on this topic, see How to Create a Salesforce Action Plan at: <https://bit.ly/3t1Tfix>

Once the scope of the sprint is decided, items should not be added unless necessary. The product owner or scrum master’s responsibility is to ensure there is no unnecessary scope creep. In only critical circumstances should items be added to an ongoing sprint.

**Tips for Success**

Often organizations start with an informal process for requesting Salesforce changes (email requests, instant messaging a Salesforce administrator, etc.). However, this is not sustainable long-term and will often lead to overlapping functionality and a slower velocity of changes made.

For optimal results, a standardized process must be documented, followed, and be championed by the leadership team. The process should include these essential steps:

* Make formal requirements gathering document
* Prioritize the requirements in a project management tool
* Judge the effort needed for all requirements
* Have a planning session to determine what will be worked on during the next sprint
* Work on a set of requirements and only those requirements for each sprint
* Inform all stakeholders of what is being worked on in each sprint

## Business Requirements Document Template

The following table can be used as a starting point for the Business Requirements document.

|  |  |
| --- | --- |
| **Requestor Name:** |  |
| **Requestor Department:** |  |
| **Date of Request:** |  |
| **Type of Request:** *(Bug, Enhancement, Question, Maintenance)* |  |
| **Description of Request:** *(Note pain point with the current process, if applicable)* |  |
| **Current Functionality:** |  |
| **Business Value:** |  |
| **Business Impact:** *(High, Medium, Low)* |  |
| **Business Impact Justification:** |  |
| **Current Workarounds:** *(If applicable)* |  |

## Salesforce Sandboxes

Insert company’s Salesforce subscription comes with the ability to create “Full” and “Partial” sandboxes that are recreations of the full production instance of Salesforce, but with either a complete “Full” or partial “Partial” replica of data. As a result, these two types of sandboxes will contain data about customers and prospects.

A license for Salesforce Data Mask can be purchased to anonymize sandbox data. For more information on this topic, see Salesforce’s documentation at: <https://sforce.co/30PckKy>

Salesforce allows for the refreshing of sandboxes, which wipes all existing data in a Full or Partial sandbox and creates a new version that replicates what is in the production instance.   
At that time, any data that was deleted from the production instance will cease to exist in the sandbox.

Insert company’s sandbox refresh policy is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Use** | **Refresh** |
| Production | Insert myDomain Name | • Live Environment |  |
| Full Sandbox | Insert Full Sandbox Name(s) | • User Acceptance Testing  • Staging  • Performance Testing | Insert full sandbox refresh period (begin with 6 months and reduce) |
| Partial Copy Sandbox | Insert Partial Sandbox Name(s) | • Training  • ‘hotfix’ testing | 30 days |
| Developer Pro Sandbox | Insert Developer Pro Sandbox Name(s) | • Merge changes from all   developer sandboxes  • Integration Testing | After every major release |
| Developer Sandbox | Insert All Developer Sandboxes currently in use | • Development and testing in   an isolated environment | After every major release |

A best practice we share with our clients is to use a partial copy sandbox as part of the company’s deployment strategy. Having a partial copy sandbox that is untouched and always kept in sync with production provides an environment with a good amount of data and the same metadata configuration as production. This is an ideal environment to test ‘hotfixes’ before they get pushed to production.

**What is a ‘hotfix’?**  
  
It’s a bug or issue found by users from the latest release that can’t wait for the next release to be fixed and therefore require immediate attention.

These are often found in the middle of the next release cycle, meaning the sandboxes are out of sync with production. However, developers can work on the fixes in the partial copy sandbox, as it is a metadata replica of production and not have to worry about any conflicts from the current sprint.

## Release & Meeting Cadence

A deployment will happen every other week, insert day and time here. A bi-weekly deployment meeting will be held to mitigate the business impact of unforeseen errors or issues during a release. It is the responsibility of the insert job title to communicate the deployment window to all users and ensure all required stakeholders have the appropriate understanding of the scope of each deployment. Deployments will occur on Tuesdays, Wednesdays, or Thursdays to allow for resources to be available to help troubleshoot if issues arise.

An essential requirement of a strong release management strategy is establishing a release cadence. The release cadence should have a set of meetings that ensure all required stakeholders have the appropriate understanding of the scope of each deployment. It is best to do deployments in the middle of the week and not on Fridays as issues may come up during the weekend and there will be limited resources to help troubleshoot.

To support the release cadence, a few meetings and artifacts are needed:

* A log of all the proposed changes will be maintained
* Two business days before deployment, there will be a meeting to discuss the ready changes, any dependencies, and manual steps that need to be undertaken
* After this meeting, the log would be submitted to insert Salesforce Deployment Lead’s Name and no new items can be added to the release’s deployment after this. All those who have access to make changes to the system must be notified of all the components in the deployment.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **M** | **TU** | **W** | **TH** | **F** | **M** | **TU** | **W** | **TH** | **F** | **M** | **TU** |
| Build |  |  |  |  |  |  |  |  |  |  |  |  |
| Deployment to Merge  (as build completes) |  |  |  |  |  |  |  |  |  |  |  |  |
| Unit Testing/QA in Merge |  |  |  |  |  |  |  |  |  |  |  |  |
| Deployment to Full Sandbox |  |  |  |  |  |  |  |  |  |  |  |  |
| Performance and Regression Testing |  |  |  |  |  |  |  |  |  |  |  |  |
| User Acceptance Testing |  |  |  |  |  |  |  |  |  |  |  |  |
| Deployment Meeting |  |  |  |  |  |  |  |  |  |  |  |  |
| Sign-off on log |  |  |  |  |  |  |  |  |  |  |  |  |
| Deployment to production |  | **P** |  |  |  |  |  |  |  |  |  |  |

The figure above is a schedule of a two-week sprint where:

* The build is scheduled to be completed by the end of the first week.
* The second week is spent on deployments to the full copy sandbox, testing, and bug fixes related to the development.

If there are issues that arise during performance testing and UAT, the developers will correct these issues in their development sandboxes, promote the changes as per the standard process and re-submit the functionality for UAT until it passes.

### User Acceptance Testing (UAT)

At minimum, a UAT template should contain: the steps users will conduct, the acceptance criteria (expected results) for each scenario, a column for users to indicate the results and a column for any developer or tester notes. An example UAT template can be found [here](https://docs.google.com/spreadsheets/d/1UUxkeBjE_nvdvluBc-AQNk7Ojg8V_P0V7Rpr_RP837Q/edit#gid=0).

End-users will test the functionality in a sandbox environment before it is deployed to production. Users will be given a document that outlines the steps they need to execute and will provide any feedback regarding the test cases through this document.

### Communication Plan

As part of the deployment process, insert company name will announce a maintenance window well in advance in which users will be unable to access the system.

Additionally, if required, insert company name will schedule training with the appropriate users to prepare them for new functionality.

### Release Notes

To be effective, release notes need to be written in an easy to understand way. Release notes for internal users can consist of text, screenshots of the system and links to video recordings demonstrating new functionality.

Release notes are an effective way to communicate changes that have been made to the system.

To conclude the sprint, insert company name will create a release notes document containing any bug fixes or functionality updates.

# Updates

This policy will be reviewed and updated at a minimum, on an annual basis by the insert company name Salesforce team.

# Exception Requests

Insert reference to company name policy on this.

# Approved Exceptions

This would be updated when/where exception requests are approved.

|  |  |  |  |
| --- | --- | --- | --- |
| **Exception Item** | **Requested By** | **Approved By** | **Approved Date** |
|  |  |  |  |
|  |  |  |  |

**FINAL THOUGHTS FROM CLOUDKETTLE:**

We hope this template provided your team with the outline and critical elements   
to create your own Salesforce Release Management process. If you have any questions about this template or Salesforce Sales Cloud optimization, reach out today.

[REQUEST A FREE CONSULTATION](https://www.cloudkettle.com/contact-us/)  
We love helping enterprise organizations transform Salesforce to drive revenue.

