How to Create an Implementation Plan

On how to create an implementation plan.

Introduction

An implementation plan is used during the deployment of an application to client test and production environments. It is used by the developers who are doing the deployments as well as by the testers who would be testing the deployments.

It provides steps on what versions of the application to deploy and roll-back, any related database and other scripts to be used for the deployments, any backout steps in case of an issue, dates and times on when they should occur as well as which user is responsible for doing what. There is also information on any escalation contacts in case of any serious issues.

Usually a change request (CR) and implementation plan are created at the same time, after the application has passed regression and user acceptance testing (UAT), prior to the application being deployed into client test and production environments.

Creating an Implementation Plan

The easiest way to create an implementation plan is to duplicate an existing one in Confluence. Open the existing implementation plan in a web browser and click on the '...' on the top right corner.

From the menu select 'Copy' and from the 'Copy page' dialogue that appears, make sure that 'Space' is selected as 'Application Development', 'Parent page' is selected as 'Implementation Plan Guidelines' and click on the 'Copy' button. If an existing implementation plan is used to copy from then these values should already be correctly selected.

It will make a copy of the original implementation plan and go into edit mode. The title should be edited by removing the 'Copy of' text so that only 'Implementation Plan for ' text remains, the correct change request (CR) Jira ticket number related to this implementation plan should be used along with a description of the main application that is being deployed.

References to the environments that the application is being deployed to should also be mentioned, 'xxx' for client test and 'xxx' for production, e.g.:

xxx

See xxx

Modifying the Implementation Plan

In the 'Introduction' section, there should be a link to a change request (CR) Jira ticket. This should be updated to link to the change request related to this implementation plan. Click on the Jira ticket icon in the text and from the menu that appears, click on 'Edit'. There is a text box at the top of the 'Insert JIRA Issue/Filter' dialogue that appears and below this text box is the text 'Search using any issue key, search URL, JIRA link, JQL, plain text or filter'.

In this text box there should be a value similar to 'key = CR-xxx'. This value should be changed so that the 'CR' number is the one from the change request Jira ticket that is related to this implementation plan. Click on the search button on the right of this text box and it will search for this ticket in Jira. When it has found the matching record, make sure that the checkbox next to it is checked and press the 'Insert' button so that the Jira link updates.

Below this there should be a ‘Change History’ Jira macro that displays the document’s change history, when it is not being edited. This should be there if the document was copied from an implementation plan that already had this macro in it.

Otherwise it can be manually added. Click on the ‘+’ on the top right and select ‘Other macros’ from the menu. There is a search text box towards the top right of the ‘Select macro’ dialogue that appears with the title ‘Search’ in it. In this text box enter the word ‘history’ and it will filter the macros below so that the ‘Change History’ macro becomes visible. Click on it and press the ‘Insert’ button.

Modifying the Pre-Implementation Tasks

The dates and times of when this feature would be deployed to client test and production environments should be known prior to doing this.

The section titled ‘Run List’ or ‘Implementation plan’ has a table with various sections in it. This table contains the dates, times, task details including any database and any other scripts that need to be run and who is responsible for the tasks on the day of deployment into client test and production environments.

The section titled ‘PRE–IMPLEMENTATION TASKS’ contains information about the tasks that need to be performed prior to the main deployment task. These are preparatory tasks that are done prior to performing the main tasks. An example is stopping a service prior to performing the main deployment tasks.

Modifying the Implementation Tasks

‘IMPLEMENTATION TASKS’ contain the main tasks of the deployment into client test and production environments. The first part of this section covers the deployment tasks into the client test environment. The first tasks could be the running of database scripts.

To enter these details, modify the first row of this table section. In the ‘Date’ column, click on the date picker and pick the date of deployment into the client test environment. In the ‘Time’ column modify the time of deployment. In the ‘Task’ column give a summary of the deployment tasks.

In the ‘Instructions/Detail’ column, give the exact details of the changes that are required to be done. In the ‘Who’ column, enter the user who is responsible for this task. Clicking on the cell and entering the ‘@’ character along with name of the user will show a list of suggestions from which the correct user can be selected.

The ‘Status’ column would be changed to ‘DONE’ by the user who performs this task on the day of deployment, so it should be left blank. Here is an example:

IMPLEMENTATION TASKS					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	05:00 pm	Deploy scripts to xxx database	Execute the attached two SQL scripts xxx.sql and xxx.sql on the servers: xxx xxx	xxx	

The is task is for deploying or running database scripts on the 09th of June 2022 at 05:00pm. The responsible user is xxx. The ‘Instructions/Detail’ mentions the names of the attached scripts

that should be run along with the names of the database servers that they should be run on. ‘xxx’ refers to the client test database and ‘xxx’ refers to partner test.

Scripts and other related files can be attached to the implementation plan by clicking on the top right ‘...’ and selecting ‘Attachments (xx)’ link, where xx represents the number of attachments that have already been attached. This cannot be done while the document is being edited but can be done once it has been published.

There can be tasks for deploying from Octopus Deploy, e.g.:

IMPLEMENTATION TASKS					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	05:00 pm	Deploy xxx	Deploy from Octopus the following version into xxx: xxx xxx	xxx	

This task is for deploying from Octopus Deploy on the 09th of June 2022 at 05:00pm. The responsible user is xxx. The environment to deploy to is ‘xxx’, which is the client test environment. The application to deploy from Octopus Deploy is ‘xxx’ and the version to deploy is 1.0.68.0 (build or release number 68). ‘xxx’ is synonymous with ‘xxx’, which is in the title of the document.

The link is to the release in Octopus Deploy. This link can be obtained by going to the deployment details page of the application in Octopus Deploy using the web browser, in this instance it is ‘xxx’. On this page there is a ‘Release’ column with a list of release version links for the application. Right click on the required release version link and select ‘Copy link address’ which would copy the link to that release. This link can now be pasted into the implementation plan as shown above.

This version for client test and production would be the same version that was deployed from Octopus Deploy to regression and user acceptance testing, after the feature branch was merged with the master branch. If there were no issues during regression and user acceptance testing then this release is the one to be deployed into client testing and production environments. This is the version of the application known as the release candidate.

The same above implementation tasks are repeated for the production environment as well but at a different time. It is always after the deployment into the client test environment and after product verification testing has been completed in that environment, e.g.:

IMPLEMENTATION TASKS					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	06:00 pm	Deploy scripts to xxx database	Execute the attached two SQL scripts xxx.sql and xxx.sql on the servers: xxx	xxx	

This is the same database script deployment as for client test, but the only differences are the deployment time and the database name that the scripts should be run on, which in this case is the production environment database ‘xxx’.

The same Octopus Deployment task is repeated for production, but at a later time, e.g.:

IMPLEMENTATION TASKS					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	06:00 pm	Deploy xxx	Deploy from Octopus the following version into xxx: xxx xxx	xxx	

It is exactly the same version of the application as for the client test environment, i.e. the release candidate, but it is this time being deployed to 'xxx', which is the production environment.

Modifying the Post-Implementation Tasks

This is the section titled 'POST IMPLEMENTATION TASKS'. The post implementation tasks usually involve product verification testing (PVT), e.g.:

POST IMPLEMENTATION TASKS					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	05:20 pm	PVT-xxx	PVT for xxx	xxx	
09 Jun 2022	06:20 pm	PVT-xx	PVT for xxx	xxx	

Notice how this testing is scheduled for 20 minutes after the deployments into client testing and production environments, which were scheduled to occur at 05:00pm and 06:00pm respectively previously. This time gap between deployment and the testing would need to be increased for larger deployments.

Modifying the Backout Tasks

This is the section titled 'BACK OUT TASKS (ONLY IF REQUIRED)'. These tasks are only performed when unexpected issues arise. This can occur during deployment into client test or production environments or during product verification testing in either of these environments. Then the deployments need to be rolled back to the previous versions. This is why the phrase 'ONLY IF REQUIRED' appears in the title, e.g.:

BACK OUT TASKS (ONLY IF REQUIRED)					
Date	**Time**	**Task**	**Instructions/Detail**	**Who**	**Status**
09 Jun 2022	05:30 pm	Deploy xxx	Deploy from Octopus the following version into xxx: xxx xxx	xxx	

09 Jun 2022	06:30 pm	Deploy Backoffice AU	Deploy from Octopus the following version into xxx: xxx xxx	xxx	

The time for the roll-back operations are scheduled 30 minutes after each deployment of build or release number 68 into client test and production environments and 10 minutes after product verification testing. This time gap between deployment, testing and rolling back, would need to be increased for larger deployments.

Both the ‘xxx’ and ‘xxx’ environments are scheduled to be rolled back to the previous build or release number 64 in Octopus Deploy.

Updating the Other Information

The ‘Communication Plan’ table contains the details of the users who would be deploying the application into client test and production environments and testing it. The ‘Escalation Contacts’ table contains details of users to contact in a serious or emergency situation.

Publishing the Document on Confluence

Once the implementation plan changes are ready, click on the ‘Preview’ button on the bottom right to preview them. Review and make any required modifications by clicking on the ‘Edit’ button. When the changes are ready to be published on Confluence, click on the ‘Publish’ button.

The document will get published on Confluence. Now the ‘Change History’ macro at the top of the page will show the first published version information of the document in Confluence.

Scripts and other related files can now be attached to the implementation plan by clicking on the top right ‘...’ and selecting ‘Attachments (xx)’ link, where xx represents the number of attachments that have already been attached.

To attach a file, click on the ‘Choose File’ button, browse to a file, select it and click on the ‘Open’ button. Add a comment in the ‘Comment’ field if required. Then click on the ‘Attach’ button. More file can be uploaded at the same time by clicking on the ‘Attach more files’ link.

Updating the Change Request (CR)

Once the implementation plan is finalised and published on Confluence, the related change request also needs to be updated. From the web browser, copy the URL link to the newly created implementation plan from the web browser address bar.

Open the change request related to this implementation plan in the web browser and go to the ‘Implementation / Backout Plan’ tab in the top section. Click on the value next to the ‘Implementation / Backout Plan’ label to edit it, enter the URL of the implementation plan there and click on the tick to save changes. Other fields can be similarly edited and saved.

In the left middle, there is the “Issue Links” section. The link to the implementation plan can also be added to this section by clicking on the ‘+’ to the right and filling in the ‘Link’ form. In this form choose ‘Confluence Page’ on the left to link to the implementation plan. Paste the copied URL to the implementation plan in the ‘Page URL’ field and add a comment if required. Click on the ‘Link’ button to create the link.

The ‘Dates’ values on the bottom right including ‘Start Time’, ‘End Time’ and ‘Backout Start Time’ need to be updated to reflect the values in the implementation plan. The ‘Start Time’ is the start of deployment into the production environment, in the above example from the implementation plan it is 06:00pm, so this should also be set to the same value.

The ‘Backout Start Time’ is the scheduled roll-back time for the production environment, which is 06:30pm from the implementation plan example above, so this value should also be set to the same value.

The ‘End Time’ is 07:00pm, which is when the production environment deployment and product verification testing is scheduled to finish. The date of these three fields should be set to 09th of Jun 2022, which is the same date in the implementation plan example above.

Now the implementation plan is ready to be reviewed by another developer.

See xxx

Meeting Recording

The was a meeting on the 07th of June 2022 and this document refers to that meeting. The recording of this meeting is available at this network location:

xxx

It is in the folder named “2022-06-07 TeamCity, Octopus Deploy, Change Requests and Implementation Plans” at the above network location.