**Overview*of XL Release :***

XL Release is an enterprise release coordination software solution that allows you to:

* Plan, track, and execute release plans from code drop to end user
* Proactively avoid release delays and failures by tracking pending tasks, risks, and dependencies
* Accelerate your delivery process by replacing manual with automated tasks and standardizing release plans

XL Release helps you deliver higher quality software faster by providing a single source of truth, increasing the level of automation, and standardizing the release process.

XL Release is for everyone who needs to work on, manage, or report on a release.

**Releases** are at the heart of XL Release. A release represents a number of activities in a certain time period, with people working on them. XL Release allows you to plan, track, and execute releases automatically. It acts as a single source of truth for everyone who is involved in making the release a success.

A release is divided into **phases**, which represent logical stages in the process that must happen in succession. For example, a release could include Development, QA, and Deployment phases. In XL Release, a phase is a grouping of tasks, which are the activities that must be done to fulfill the release.

**Tasks** are activities in a release. In XL Release, everything that must be done is defined as a task. There are manual tasks, in which a human must do something, and automated tasks that the XL Release flow engine performs, unattended.

When a release is started, XL Release executes the **release flow**. This is the workflow of the release. XL Release determines the current task that needs to be picked up and executes it (if it is an automated task) or sends a message to the person responsible for it (if it is a manual task).

Each release has a **release owner**, the person that is responsible for correct performance of the release. If something goes wrong, the release owner will be notified. For example if an automated task produces an error, or one of the people working on a task indicates that he is in trouble.

A **template** is a blueprint for a release. You can use a template to start different releases that have the same flow. A template is very similar to a release; however, some functionality is different because a template is never executed directly. For example, there are no start or end dates in a template; most tasks are assigned to teams rather than actual people; and **variables** are used as placeholders for information that differs from release to release, such as an application's version number.

Each release or release template defines a set of **teams**. A team is a logical grouping of people who perform a certain role. For example, on a release you can define a Development team, a QA team, an OPS team, and a Release Management team.

***Problem Statement :***

XL Release is configured more than 180 templates for 70 different projects across RWS. The variables that are used in the xl release templates are place holders and are supposed to be used be filled up with values when the release is created. XL Release invokes XL Deploy to perform the deployment to target severs. All the templates are developed with the same set of variables to maintain consistency.

There are some variables that are present in XL Release that are common for all the 180+ templates and they need to be modified at once. For example , XL Deploy requires an admin user id and password to deploy the artifacts and we cannot have a non user account due to IRM team's restrictions. ***The password for an user id will expire for every 90 days and that needs to be updated manually across all the 180+ templates with is a rudimentary manual job taking at least 3 to 4 hours to update and verify the template.  There are many other cases like updating email distribution list where the same variable has to be modified across multiple templates. These tasks of creating and updating variables  can be automated to save time and effort.***

***Solution:***

XL Release Automator is a internal tool created to reduce the manual effort involved in creating/updating XL Template variables.

XL Release works on top of API calls. Each button click calls an API and paints the data on the XL Release dashboard. Instead of updating the data manually on the XL Release screen , the API call can update the same in the XL Release database.

Each XL Release base folder would contain multiple sub folders and each sub folder would contain multiple XL Release templates.

Each XL Release template would contain variables. All the variables can be accessed using the XL Release API call using basic authentication.

So a console application is developed that uses the XL Release API to pull all the XL Release templates as per the settings in the configuration file (rootfolders.xml file) to create and update the variables as set in the variables.xml file.

All the calls for the xl release server can be made to the above URL with a basic authentication header with a base64 authentication key.

The App.config file has to be updated with the base64 version of the authentication key which is the base64 hash of name:password.