

NASA & Octopus Deploy



"Implementing Octopus Deploy into our software delivery lifecycle has significantly reduced the complexity, errors, and man-power required to deploy applications."

- Michael Knopf, Software Architect

Their Story

Under the direction of NASA IT-C2 we provide information management and communication services in support of space launch operations and related activities. With nearly 400 production applications, the vast majority being custom built by our software development organization, we play a major role in supporting NASA's current and future missions at Kennedy Space Center, Florida USA.

How did you deploy software before using Octopus?

Our legacy product was a custom built installer program which requires a lot of manual input configuration, essentially providing a wizard interface for manual deployment of applications to target environments. The configuration data was stored in Microsoft Word documents on a SharePoint instance, often resulting in outdated/lost/incomplete information as well as copy/paste errors during the deployment process.

How has using Octopus improved the way you deploy software?

We went from a total of 50 manual steps for deploying to production down to less than 20, completely eliminating the need for our System Administrators to be involved in the process, all while improving our deployment process so that it's automated, standardized, and repeatable.

What are the main benefits for using Octopus in your organization?

Just some of the benefits are (this is the short list):

- Elimination of storing configuration info in Word documents on a file share, now everything is stored in the variables within Octopus Deploy.
- No more copy/paste of anything during a deployment
- Elimination of nearly all paperwork aspects required to request a deployment (HUGE time saver)

What are the main benefits for using Octopus in your organization? (continued..)

- Huge increase in the speed in which we can promote between environments
- Dramatic increase in overall security (no more DB passwords and other sensitive information stored in source control or documents)
- Automated encryption of the sensitive sections of config files such as Connection Strings
- True enforcement of workflow to promote an application between environments (you will not be able to promote from DEV to PROD without going to TEST first, every part of the workflow is enforced and there is no way around it)
- Ability to block releases that contain issues so that it cannot be deployed to the next environment in the lifecycle
- A snapshot of the release process and variables used for that release are taken when the release is created, this keeps everyone honest, if the variables were changed after the release is created those changes are ignored (you cannot sneak in any last minute changes without it being tracked via the Audit Trail)
- A complete audit history of changes to the deployment plan, release, and system as a whole is maintained. You can even see what was changed, added, or removed
- Automated emails are generated informing the appropriate people/teams when a deployment has started, finished, failed, or is awaiting approval
- Release Notes are automatically generated when the Build server kicks off an auto-release to the DEV environment, these notes contain links to the Source Code for the release and to the specific Build in Build server, these release notes are preserved as the release is promoted to TEST and finally on to PROD. This truly ties all these systems together for historical tracking.
- Version numbers for the release are automatically generated and fully standardized, no more "guessing" what the version of the application should be.
- A complete Log of the deployment is maintained and available in the release details, this will become especially important in determining why a deployment failed
- Sensitive information is secured, such as API Keys, Database Connection string, etc.... and encrypted in Octopus Deploy.
- During the release the existing application in the target environment is automatically backed up, stored using a standardized and automated naming convention and location (no more saving it as a zip file on your desktop).
- Automated Rollback of failed deployments, if a deployment fails or is aborted the backup is automatically restored.