



# Finding the Return on Investment in DeployHub

How to calculate DeployHub ROI

## Table of Contents

Table of Contents .....	2
Summary .....	3
ROI Formula .....	3
DeployHub Technology Monetary Benefits .....	4
Calculate Your ROI .....	5
. Standard Cost of DeployHub Implementation .....	5
Savings Per Application Team Per Year .....	6
Example ROI for DeployHub .....	6
Conclusion .....	7
About OpenMake Software .....	7
Tracy Ragan – COO and Co-Founder, OpenMake Software .....	7

## Summary

This paper is intended to serve as a Return on Investment (ROI) calculation tool for organizations looking to invest in Release automation software, particularly DeployHub. It is a guide to assist you in understanding where the maximum return on investment can be gained through package and deploy automation.

Generally you would calculate monetary, technology and business benefits to determine the ROI for any technology. For simplicity, we have only included the monetary benefits for this ROI calculation. The ROI numbers will improve if less tangible benefits are calculated into the equation. Less tangible benefits would include technology and business benefits, for example cost saving associated to a reduction in production 'down times' or the cost savings in improved collaboration between development, testing and production.

## ROI Formula

Return on investment (ROI) is calculated as a ratio of net benefits over costs reported as a percentage. This formula is represented as follows:

$$\text{ROI} = [(\text{Monetary Benefits (Tangible and Intangible)} - \text{Cost of using Release Automation Technology}) / \text{Cost of using Release Automation Technology}] \times 100$$

For example, if an organization determines that increased automation can result in a 20% increase in efficiency resulting in a net savings of \$300,000 per year for 20 developers and the cost of the investment was \$140,000 then the ROI for the first year would be calculated as follows:

$$\text{ROI} = (\$300,000 - \$140,000) / \$140,000 \times 100 = 114\%$$

This equals a return of \$1.14 for every \$1.00 spent on the new technology in the first year.

## DeployHub Technology Monetary Benefits

The ROI for DeployHub can be easily calculated based on two monetary saving features: the elimination of one-off deploy scripts and the elimination of end target agents.

Release Automation tools are intended to provide a common interface to manage and execute deployments. Some might consider these benefits to have the highest return on investment as it improves developer productivity by increasing collaboration between developers and production control at the same time as reducing production failures. DeployHub minimizes redundant tasks and deployment errors; however its core ROI is in its ability to eliminate one-off deploy scripts and to deploy binaries without the overhead of agents. It is in these two areas that DeployHub provides the highest ROI.

The vast majority of manual scripts written by software developers is related directly to the software builds and deploys – meaning the actual process of converting source code into binaries, packaging the binaries for delivery and then moving them from point A to point B. In most organizations a large majority of scripts are dedicated to either compiling code, or packaging and deploying code. Each script supports a single application release, for a particular version, for a particular stage in the life cycle. In other words, scripts must be maintained to address the changes across the life cycle, drastically increasing the number of scripts to manage.

DeployHub reduces cost by minimizing or completely eliminating the scripts associated to packaging and deploying applications across the life cycle. Organizations can drastically reduce the number of deploy scripts and their associated cost by using DeployHub Activities to define and share deploy logic amongst all application teams, and across the life cycle to test and production..

Secondly, most organizations streamlining the release management process do so using a workflow engine such as Jenkins. The workflow engine calls on the deploy script to perform the delivery, and uses end target agents to perform the delivery. End target agents can be expensive to manage and maintain, particularly in production environments where hundreds (and sometimes thousands) of end targets are required. DeployHub eliminates the need for end target agents and the associated cost of maintaining the end target agents across development, testing and production server environments. DeployHub can be called by the continuous integration server, or can manage the deployment workflow independent of a CI process with on-demand or scheduled releases.

## Calculate Your ROI

To calculate the ROI gained through DeployHub, give a reasonable estimate for the time it takes to create a scripted deploy process for a single project. Then assign another 30% for a team's Release Manager for ongoing maintenance. You can estimate that a single full time resource would be devoted to developing a scripted deploy process for 2 to 6 months, and then 30% of that person's time for the life of the application.

Example: 1 Full time resource - \$60 per hour for 3 months (average) = \$28,800  
 Annual Maintenance cost of manual package and deploy scripts (\$720 per week)  
 \$34,560

**The cost of manual deploy scripts per application team - year one \$63,360**

**The cost of manual deploy scripts per application team - year two \$34,560**

Now, calculate your ROI for addressing the maintenance of your end target agents. There are four cost associated to managing agents. Agent license, installation cost, maintenance and upgrade cost and the cost associated to deployment failures due to unavailable agents.

For example, considering a 'free' Jenkins agent that must be installed on each Server is estimated to be 2 hours per Server – or 20 hours for an environment of 10 servers. . Maintenance and upgrades can be calculated similarly. A failed deployment to do an unavailable agent can require 2-4 hours per failure.

Example: 1 Full time resource - \$60 per hour for 20 hours = \$1,200

Yearly Maintenance (3 times per year) = \$3,600

An Agent Failure once per week would cost \$60 per hour for 3 hours multiplied by the frequency of your deployments in development, test and production. In our example we will estimate one failure per week. (\$180 x 52 deployments) = \$9,360

**Over one year, the cost savings would be \$14,160**

Total cost year per year for a one-off scripted solution using an agent based workflow engine such as Jenkins:

Script Development	\$28,800
Script Maintenance	\$34,560
End Target Agent Maintenance	\$14,160
<b>Total</b>	<b>Year One - \$77,520</b>
	<b>Subsequent Years \$38,720</b>

## Standard Cost of DeployHub Implementation

In keeping with our theme of a **single** application team with a production environment of 10 end targets and 10 DeployHub users, the following costs are associated with purchasing and implementing DeployHub:

### Year 1

Software License Fees	\$1,000
Annual Support	\$250 (20% of list price)
On going administration	1 FTE - 20% = \$23,040
<b>Total Cost of Technology</b>	<b>\$24,290</b>

### Subsequent Years

Software License Fees and Support	\$1,250
On going administration	1 FTE - 10% = \$11,520
<b>Total Cost of Technology</b>	<b>\$12,770</b>

## Savings Per Application Team Per Year

Year one – Scripted/Jenkins Cost – DeployHub Cost  
 $\$77,520 - \$24,290 = \$53,230$  per Application Team

Year two  
 $\$38,720 - \$12,770 = \$25,950$  per Application Team

## Example ROI for DeployHub

*\*TBD indicates that these numbers must be calculated on an individual installation basis.*

$ROI = [(Monetary\ Benefits\ (Tangible\ and\ Intangible) - Cost\ of\ using\ Release\ Automation\ Technology) / Cost\ of\ using\ Release\ Automation\ Technology] \times 100$

### Year One

$(\$77,520 - \$24,290) / \$24,290 \times 100 = 219\%$

This means that for every \$1.00 spent on technology a return of \$2.19.

### Subsequent Years

$(\$38,720 - \$12,770) / \$12,770 \times 100 = 203\%$

This means that for every \$1.00 spent on technology a return of \$2.03.

## Conclusion

When evaluating the ROI for your release automation process, it is important to understand where the largest return comes from. The largest area for improvement in the release process is addressing the high cost of redundant and error prone scripts and the high cost of managing end target agents. This is where the lion's share of cost associated to software deployments is hidden. These ROI numbers are a guideline for you to use in estimating your own ROI. These numbers can be even higher once you calculate the business benefits as well as all technology benefits that can be obtained by automating all aspects of the software release process at all stages of the life cycle and for multiple teams.

## About OpenMake Software

OpenMake® Software delivers highly reusable DevOps Solutions that allows our customers to go the 'last mile' in agile. DeployHub, our Application Release Automation solution, eliminates the friction in the continuous delivery pipeline as code moves between environments. Meister accelerates and streamlines the software compile process for highly efficient continuous builds. As a 100% self-funded organization, we have the freedom to focus on our customer's needs, delivering innovation in software builds and release. An investment in our DevOps solutions or Professional Services forms a 'technical partnership' that provides you expertise and support to solve your toughest build and release problems for today and tomorrow.

DeployHub is an Open Source project at [www.DeployHub.org](http://www.DeployHub.org).

DeployHub Pro has advanced security and auditing features. A free version is available to small teams and can be downloaded at no cost from <https://www.openmakesoftware.com/deployhub-free-downloads/>

## Tracy Ragan – COO and Co-Founder, OpenMake Software

Ms. Ragan has had extensive experience in the development and implementation of business applications. It was during her consulting experiences that Ms. Ragan recognized the lack of build and release management procedures for the distributed platform that had long been considered standard on the mainframe and UNIX. In the four years leading to the creation of OpenMake Software she worked with development teams in implementing a team-centric standardized build to release process. She can be reached at [Tracy.Ragan@OpenMakesSoftware.com](mailto:Tracy.Ragan@OpenMakesSoftware.com).



OpenMake is a registered trademarks of OpenMake Software



ASERVO Software GmbH

Konrad-Zuse-Platz 8  
81829 München / Germany

T: +49 89 7167182-40

F: +49 89 7167182-55

Email: [kontakt@aservo.com](mailto:kontakt@aservo.com)

[www.ASERVO.com](http://www.ASERVO.com)