



# How to monetize your DevOps function?





## Introduction

DevOps has largely been an internal affair for most organizations involving internal teams and internal requirements. These workflows primarily concentrate on improving or quickening the work of internal teams and are focused on cost reduction and improving efficiency instead of a revenue generating exercise. This article ponders the feasibility of extending these benefits to customers of your product while adding a new revenue stream and to monetize the DevOps function as a whole.

Despite the number of tools out there, it seems entirely possible (and profitable) to sell internal “DevOps” workflows to others who may find it useful (primarily your own customers). While the types of DevOps tools and frameworks have grown rapidly over the past decade, we see that many of them have originated from in-house tools. We’ve observed an increasing number of product companies take this approach to increase their revenue while extending an additional service to their own customers.



# When can we monetize DevOps Function?

Creating a new revenue stream is not always easy. The below list of conditions can be used as guidelines to make the decision



## Demand vs Supply

- i. Demand for the solution is a primary “go/no-go” criteria. Unless demand for such a solution exists there is no point in investing.
- ii. If there are external or third party solution providers for the problem, they are already supplying the demand. So, we have to evaluate to ensure that the demand is greater than the supply and that it is possible to beat or out-perform the other suppliers.



## Pricing and ROI

- i. Building a DevOps workflow that is both general and custom-made for each customer has its own costs. So, the question of pricing and ROI becomes very important. Estimated ROI should be significant enough to offset the cost and the project should break-even within an acceptable duration.



## Internal use and internal benefits

- i. Sometimes, more than the monetization aspect, the generalized workflow might benefit internal teams and reduce internal costs. This is another influencer for this change.



## Protection of Proprietary technology & security aspects

- i. If exposing your DevOps functions exposes some proprietary technology that we need to guard at all costs, it may not make sense to monetize the DevOps function. But we can still implement the idea internally if there are significant cost benefits.



## Examples and Use cases

Recently, some of our notable clients have taken this route and have been able to add new revenue streams by simplifying and generalizing their own internal tools

Temenos, one of the largest banking product company, recently rolled out a tool named TCDP (Temenos Continuous Delivery Portal) to some of its customers as a service (or a licensed product, if the customer chooses). Customers of Temenos are mostly banks that run a combination of products on a variety of infrastructure stacks. Along with the core product, each bank has the capability to apply a variety of customizations and enhancements. This need for customizations and additional development over the core product was an important aspect in the need for the DevOps for customers. This portal essentially is an enhanced version of their in-house deployment and testing tool and is capable of integrating with a variety of cloud platforms for deployment and testing tools to test the integration of the core product with their own customizations. The portal was built as a multi-tenant platform that supported multiple clients and users to login and run deployments on demand.

Similarly, another large client of ours, an insurance product seller from the US also built a similar eco-system for on-demand DevOps for their customers. Customers can login to a centralized portal and provision instances of the product on demand and apply their own customizations over it. We can use this instance for testing and other purposes.

For our engagement with several large product companies, we realized that this trend is going to grow. Although, this seems like an idea that works better with products, let's not rule out the usefulness of the business model for software service companies. Service companies use the generalized DevOps workflow to improve deliveries to all of their own clients.



# Execution Plan

Assuming that we've established that the idea is worthy of implementation (or at least as a proof of concept), let us see the general steps taken by most customers.

## 1. Generalize existing workflows for extendibility

- a. Internal workflows have a lot of assumptions and fixed ideas. In order to sell it to others, we need to rewrite the workflows to address custom demands and eliminate the existing assumptions. They will also have to integrate with a variety of tools, product combinations, customizations, testing tools and other frameworks

## 2. Introducing multi-tenancy

- a. As an idea, it makes a lot of sense to sell this function as a service. The service model allows making continuous improvements in the background and keeping a central control over the workflow. To be able to sell this to multiple customers as a service, it is important to think this as a multi-tenant service and can handle data and resource isolation between tenants.

## 3. Portal and Self service capabilities

- a. To be able to sell it to customers, it is important the service has a self service portal where they can login and create everything they need on the portal. This could be the most expensive component of this whole development.

## 4. Test within the organisation first

- a. Testing the new portal is very important before rolling out to customers. It might be easier to test it with internal teams that require deployments for their day to day tasks. This internal testing can produce so much feedback that can be extremely helpful in rolling out a more resilient product to customers

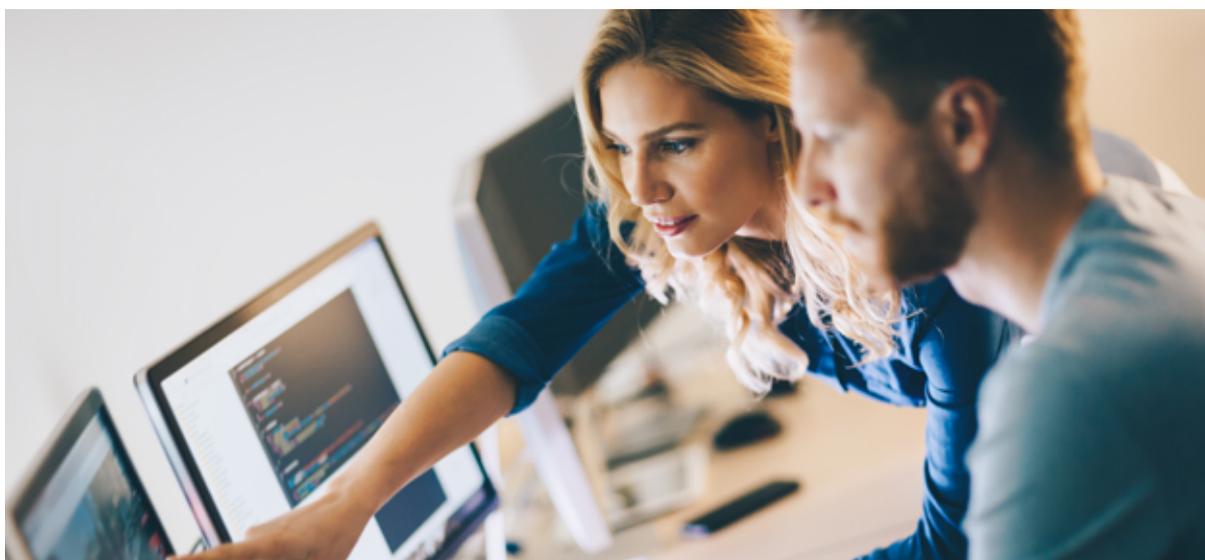
## 5. Mind Security aspects

- a. Despite internal testing, the product should be thoroughly tested for security flaws and constantly monitor for security incidents to keep fixing them,



## Conclusion

For many ISVs (Independent Software Vendors) that we have worked with, the realization that their DevOps function can be monetized came several years after other companies have already filled the demand. Although late, we did encourage them to generalize their pipelines to an extent where it can be sold. After all, the makers know a product better than anyone else.



## About Aspire



Aspire Systems is a global technology services firm serving as a trusted technology partner for our customers. We work with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in our specific areas of expertise. Our core philosophy of "Attention. Always." communicates our belief in lavishing care and attention on our customer and employees.

[For more info contact](#)

[info@aspresys.com](mailto:info@aspresys.com) or visit [www.aspiresys.com](http://www.aspiresys.com)

NORTH AMERICA  
+1 630 368 0970

POLAND  
+44 203 170 6115

INDIA  
+91 44 6740 4000

MIDDLE EAST  
+971 50 658 8831

EUROPE  
+44 203 170 6115

SINGAPORE  
+65 3163 3050