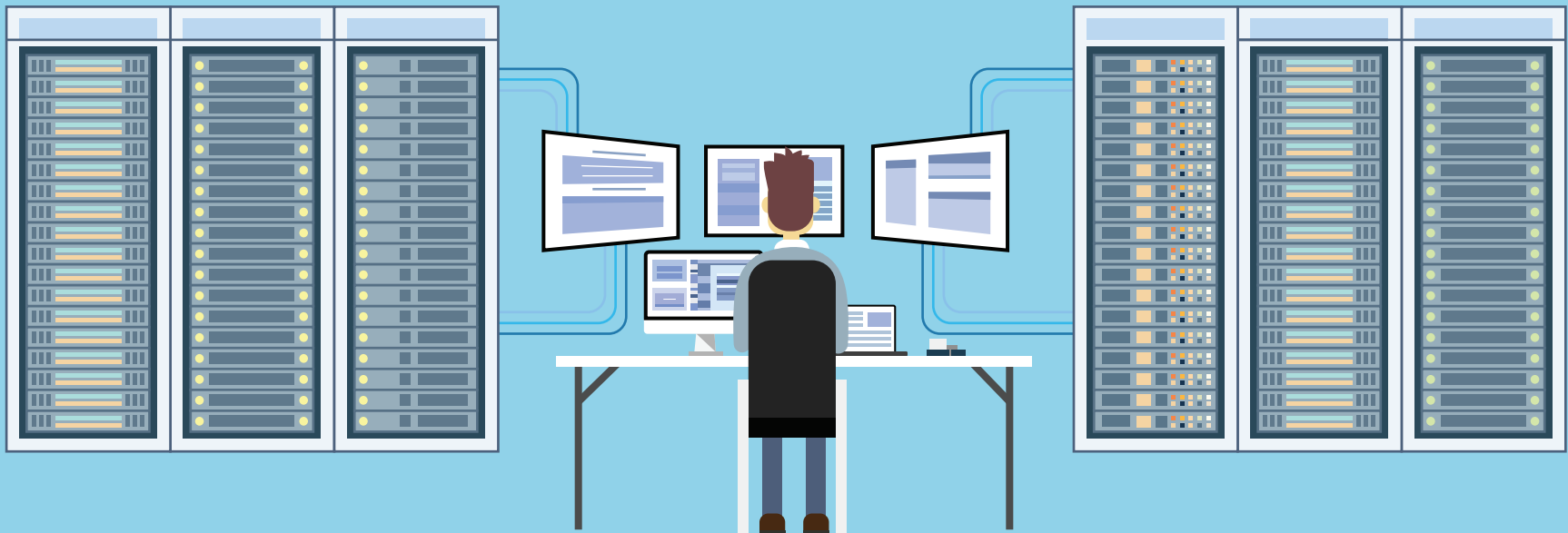


4 STEPS
TO WEBIFY YOUR
LEGACY APPLICATIONS

Hoku**Apps**

With respect to the technical aspects of legacy application modernization, every project differs, as not every company uses the same systems, infrastructures and operational environments.

But from a more holistic perspective, development and operations must execute four steps to modernize their antiquated solutions, whether they be mainframe applications or on-premise ERP deployments.



4 STEPS TO WEBIFY YOUR LEGACY APPLICATIONS

ESTABLISH A BUSINESS
CONTINUITY PLAN

4

ASSESS THE
APPLICATION

2

3

CONDUCT THREAT
MODELING

1

IDENTIFY YOUR
CURRENT AND IDEAL
INFRASTRUCTURE

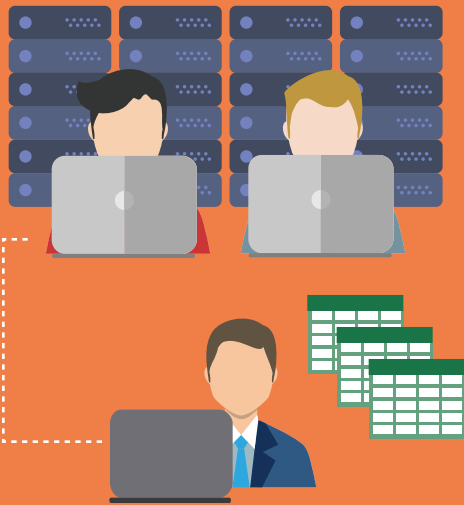


IDENTIFY YOUR CURRENT AND IDEAL INFRASTRUCTURE

01

Conceptually, there are two infrastructures:

**The one on which your
business currently
operates**



**The one on which your
organization should
operate**



For example, your customer relationship management system may not be capable of automatically logging customer web behavior data in real time, even though your developers could create a program that enables this function - operations simply doesn't allow for such a program to run.

It's possible your production environments, networks and other resources, as they run today, do not support business process optimization.

In short: figure out how your infrastructure can make the business run more efficiently. Once you understand which parameters and components will support this infrastructure, you'll be able to modernize the application accordingly.

Ask yourself:

- ➡ *Is the production environment hosted or on-premise?*
- ➡ *Should applications live within a container?*
- ➡ *Is the legacy application written in a language that makes it more difficult for you to establish compatibility between the system and the operation environment?*



ASSESS THE APPLICATION

02

Once your team has a detailed blueprint of the current infrastructure and determines how it will change over time, you can analyze the application itself to draft a modernization plan. Once your team understands the system's composition, you can execute one of the following build plans:

RE-HOSTING: Migrate the legacy application from a mainframe or in-house data center to a cloud-based environment.

This assumes the application's code is easy to integrate with your platform of choice.

CUSTOM RE-WRITE: Use a different language to develop a modern version of the legacy application, foregoing certain functions and programs that are now obsolete

This plan assumes you have the time and human capital to undertake a lengthy development project.



SERVICE ORIENTED ARCHITECTURE

ENABLEMENT: The legacy application operates as is, but exchanges data to a modernized application layer.

Of course, there are other build strategies of which you can take advantage. The type of plan you choose depends on your internal resources as well as the run environment.

Also, **TechTarget** recommended assessing how important it is to migrate legacy data into the modernized iteration. The publication suggested ranking information as high, moderate or low. For instance, information with a high rating indicates the data's unavailability would severely limit the organization's operations.



CONDUCT THREAT MODELING

03

The legacy application, its modern iteration and the run environment all have vulnerabilities. Development can allocate resources to address these vulnerabilities via the agile framework - using stories to fix bugs and the like. However, you should approach security from the objective of building it into the modernized application - a process known as threat modeling.

According to the **Open Web Application Security Project**, threat modeling is the practice of identifying, understanding and resolving the security risks associated with a system. This assessment includes analysis of not only the code itself but also the function of the application as a whole.



OWASP outlined threat modeling as a three-step process:

1

DECOMPOSE THE APPLICATION: Comprehend and document how the application behaves within the run environment, interacts with other software and assess how those interactions may open the door to hackers.

2

IDENTIFY AND RANK THREATS: Categorize dangers through STRIDE, which classifies threats based on the types of vulnerabilities hackers may use to infiltrate the application or Application Security Frame. ASF uses categories such as authorization, data protection, configuration management and other functions. This step allows you to prioritize threat mitigation.

3

DETERMINE COUNTERMEASURES: Figure out what sort of tactics would be effective in neutralizing the vulnerabilities and dangers you outlined.

How you choose to execute threat modeling is up to you.

“

For example, it may be feasible for you to break up application decomposition responsibilities by different operations and development teams based on their security knowledge.

ESTABLISH A BUSINESS CONTINUITY PLAN

04

If the operating environment encounters an unexpected outage, to which infrastructure will the application failover? Which personnel will ensure the application can access the data it requires to function properly? These are just two of many questions you should ask when developing your business continuity framework.

The Department of Homeland Security recommended conducting an operations impact analysis to determine how application downtime would affect end users. This process usually entails surveying employees who need the system to perform their duties. Once you understand the repercussions, you may conduct a gap analysis to identify your recovery capabilities' shortcomings.



Again, the manner in which you execute this application modernization plan depends on:



Time Constraints



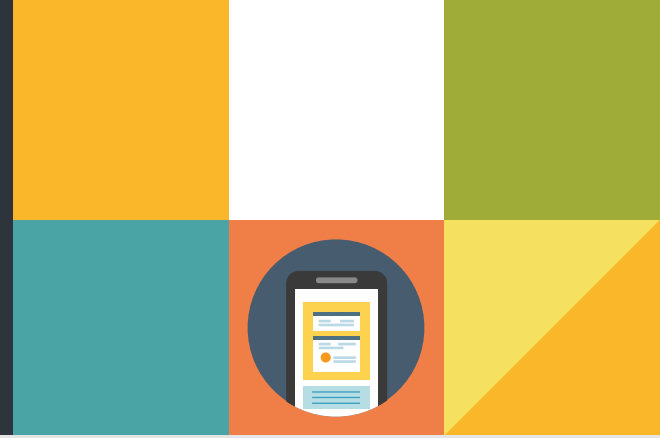
Employee Resources



Regulatory Demands

Keep in mind that one company's approach may not be feasible for your organization.

HokuApps is a leading cloud application development platform provider. Our pre-built connectors and modules provide easy legacy app integration to the cloud and empower businesses to create custom solutions based around their specific operational demands. HokuApps provides the platform and tools needed to get proprietary apps rolled out in a matter of days.



Getting to know HokuApps

The HokuApps Rapid Application Development Platform gives organizations the tools they need to turn app ideas into deployed solutions in a matter of days. It provides pre-built connectors to integrate custom apps with a variety of popular solutions, making it easier integrate legacy apps with the cloud and roll out new solutions quickly and efficiently.

Pre-built tools that drive efficiency

We have built the HokuApp platform with business value as a priority. The system is designed to empower users to create intelligent, agile solutions to fit any of their workflows. With simple user access and reliable security, you'll be able to build out apps that align with specific business needs, even for complex, data intensive processes.

Streamlined capabilities that ensure ease-of-use

Rapid app platforms alleviate the development burden on your IT teams. This is achieved through low-code systems that eliminate the need to write new lines of code for every capability.

Functionality across diverse lines-of-business

The modern enterprise is built on apps. Users need access to data and the ability to complete key tasks from any location, using any device. The HokuApps platform is built to fuel this functionality with workflows that support a variety of business units.

Aligning technology with the business

Organizations are increasingly dependent on IT systems to keep up with competitors. One-size-fits-all apps are falling out of favor as each company has unique needs. HokuApps provides the rapid development functionality you need to roll out apps that align with your specific operational demands.



For more information about HokuApps, visit www.hokuapps.com, follow us on Twitter and Facebook or reach out to us at support@hokuapps.com.

© 2016 Hoku**Apps**