



# **Rebuilding and Replatforming Legacy Applications**

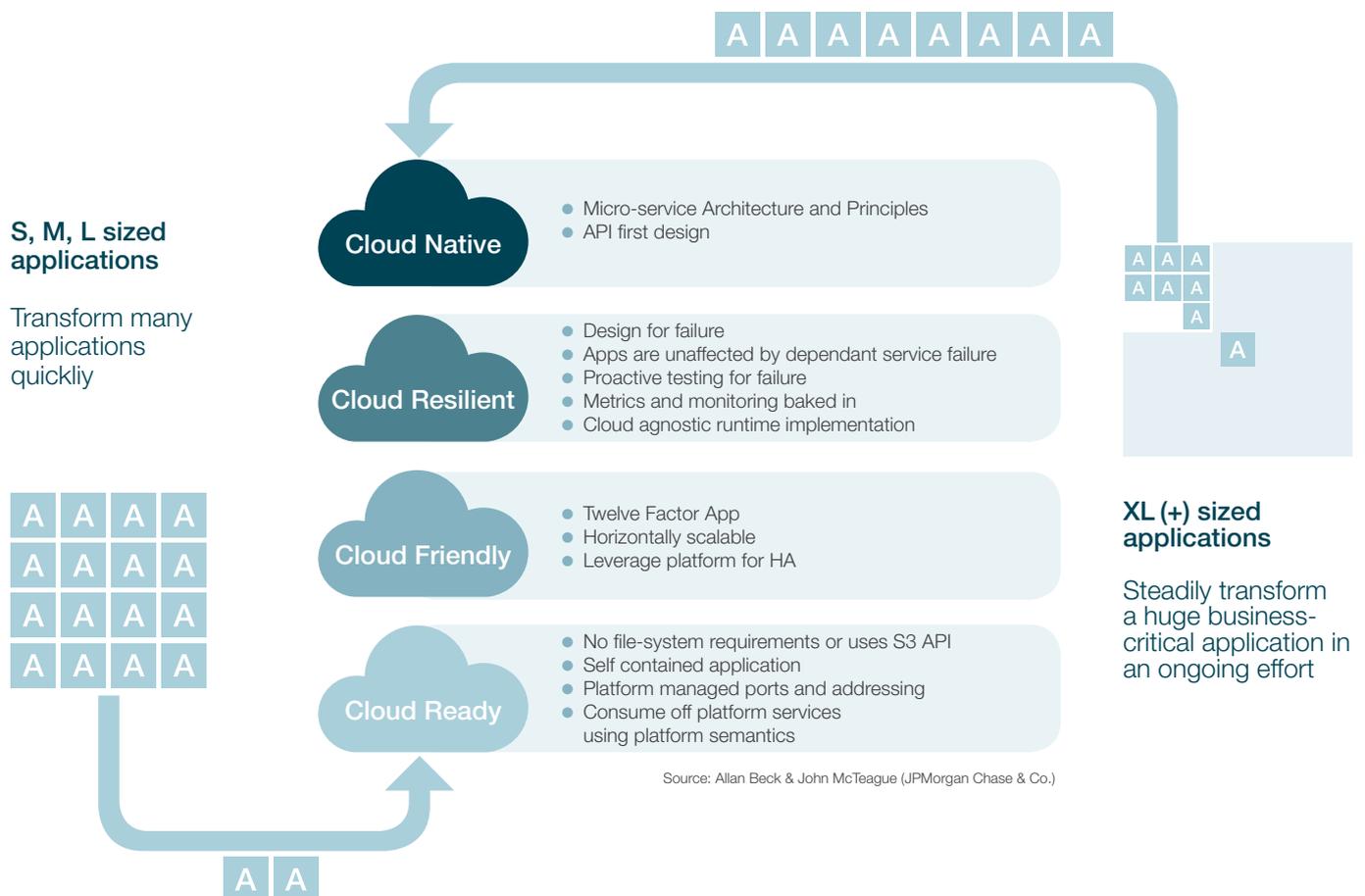
# Introduction

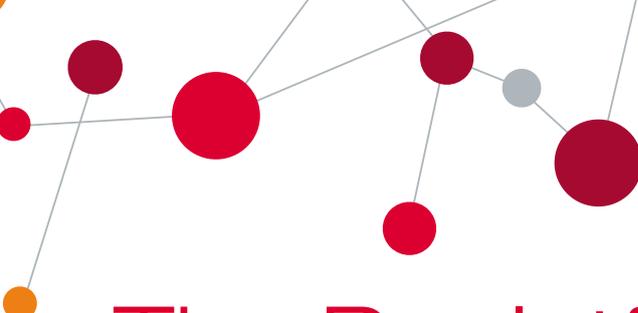
Many organizations are burdened by legacy software of many kinds, including enormous and excessively complex, monolithic applications. Study after study has shown ([www.de.cgi.com/de/cgi-insights](http://www.de.cgi.com/de/cgi-insights)) that modernizing application portfolios is a high priority for them. This whitepaper describes CGI's two-step approach for accomplishing this: (1) replatforming existing applications from bare-metal or virtual machines to containerized target environments and (2) gradually eliminating huge monolithic applications by rebuilding their functionality as native cloud services. You can greatly benefit from this if you own tens, hundreds, or even thousands of applications and need a streamlined approach for containerizing them, or if it's time for you to modernize a large business-critical backbone system that still has years of usefulness left in it but has run up against its limits in terms of manageable complexity. In the latter case, you require a partner who knows how to tackle "big ball of mud" applications – i.e., that are cobbled together from spaghetti code without a discernible architecture – and rebuild them piece by piece while applying state-of-the-art practices and principles.



# Two Approaches

We take two main approaches to modernizing applications: replatforming and rebuilding. The **replatforming approach** is depicted on the left in the figure below. It involves migrating a large number of applications running on virtual or bare-metal machines to cloud-ready containers on a cloud platform. The **rebuild approach** is shown on the right. It modernizes a large monolithic application by extracting parts of its functionality and rebuilding them as cloud-native services. We have well-defined approaches for both scenarios, along with the appropriate methodologies, technologies, and engineering skills. They are described in the following.



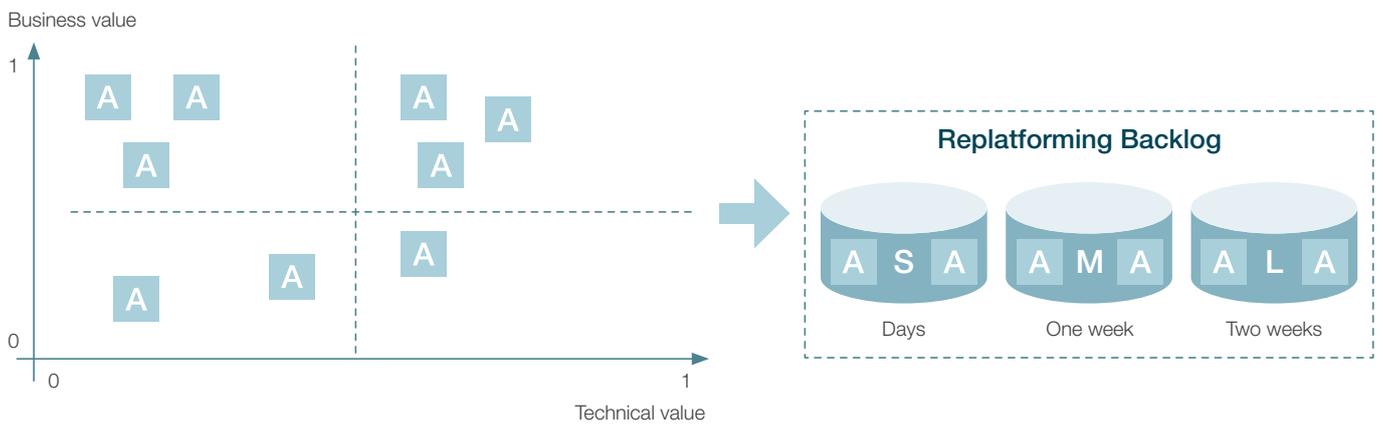


# The Replatforming Approach

In replatforming projects, we usually take the first of the two mentioned approaches: namely, migrating a large number of applications to a cloud-native runtime environment. Here the goal isn't necessarily to completely rebuild applications. Instead, they are only altered to the extent required to ensure that they will run in the target environment.

Before starting, we assess your cloud and business strategies to make sure that we're on the same page and will be migrating the right applications for the right reasons. We strive to understand your cloud strategy, your current data center setup, the level of maturity of your organization's cloud-native development practices, and various other aspects. The CGI Client Proximity Model is hugely beneficial for this. If your organization already has a relationship with CGI, it's very likely that we already possess much of the required information as well as a good understanding of your overall environment. This saves time and allows us to quickly dive into the replatforming work. Our methodology for this has components: a continual assessment stream and a continual replatforming stream.

Business value	WF	App1	App2	App3	App4	Technical value	WF	App1	App2	App3	App4
Business Criticality	0.4	8	8	8	8	Code base	0.6	3	5	3	1
Revenue generation	0.2	8	8	8	8	CI/CD	0.2	3	2	4	5
Client touchpoints	0.4	8	8	8	8	Dependencies	0.2	4	8	6	4
...	...	...	...	...	...	...	...	...	...	...	...
		5.4	5	6.2	3.8			3.2	5	3.8	2.4
<b>Business Index</b>		<b>0.18</b>	<b>0.17</b>	<b>0.21</b>	<b>0.13</b>	<b>Technical Index</b>		<b>0.11</b>	<b>0.17</b>	<b>0.13</b>	<b>0.08</b>

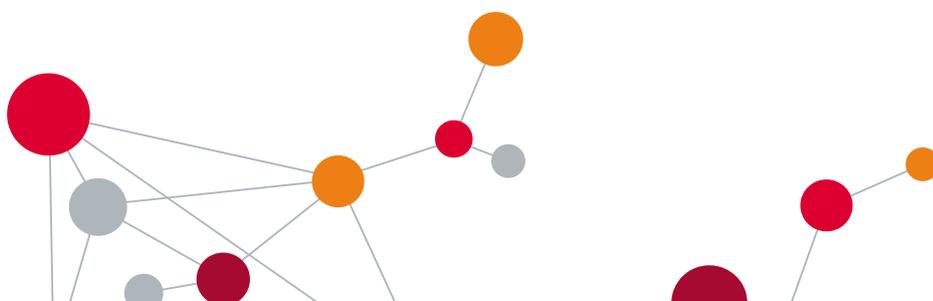


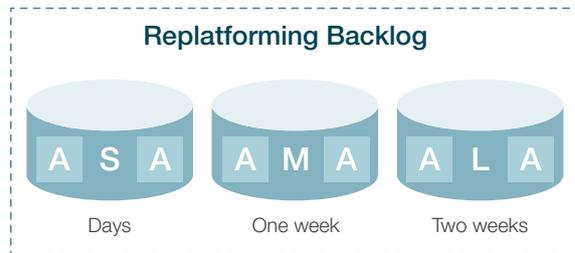
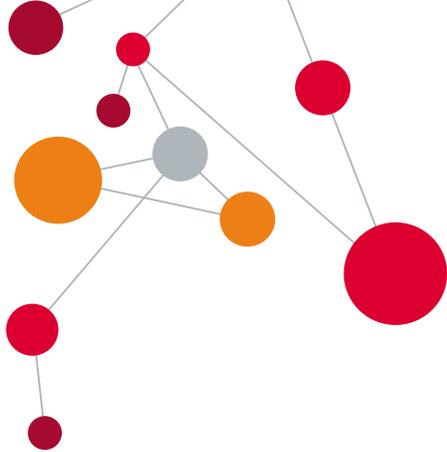


*“Replatforming. Migrate an application component to a new runtime platform. Make minimal changes to code to adapt to the new platform, but don’t change the code structure or the features and functions it provides.” – Gartner*

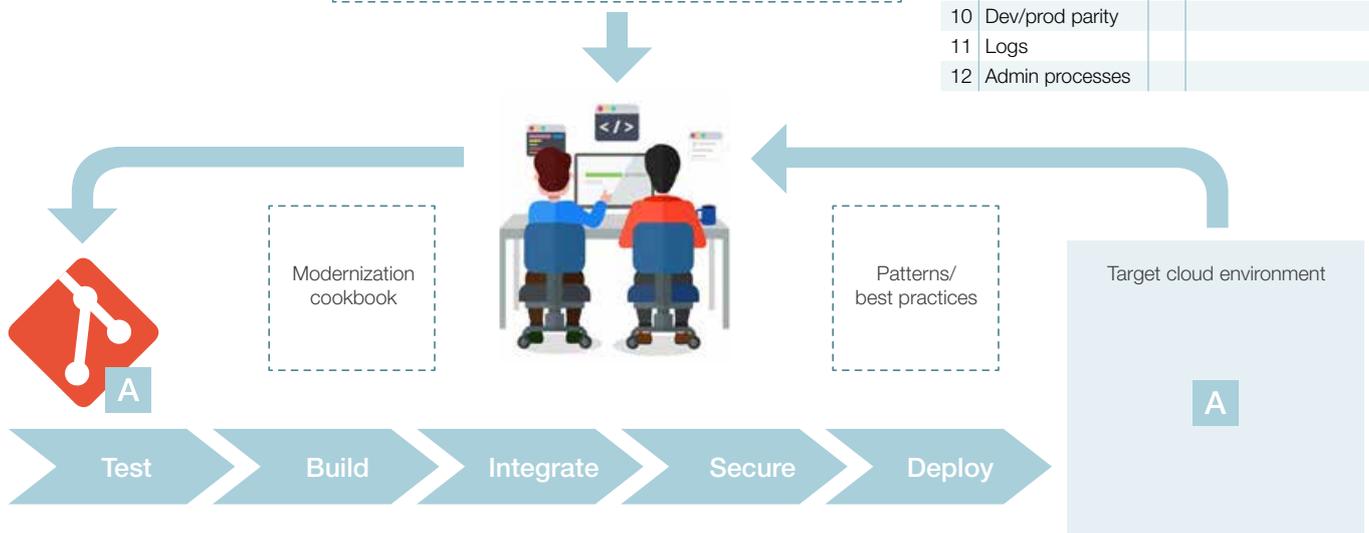
The team responsible for the continual assessment stream assesses one application at a time according to predefined criteria. The focus is on understanding each application’s business and technical value. This is accomplished with a combination of structured interview sessions and facilitated workshops in which we use state-of-the-art methodologies such as Snap analysis or event storming to gather the required information. The results of assessment are added to a replatforming backlog, where we typically group applications by labelling them as S, M and L (analogously to clothing sizes) to get an idea of how long it will take to migrate them.

Once the first applications have been analyzed and data on them have been placed in the replatforming backlog, work on the continual replatforming stream can begin. A group of experienced engineers takes applications out of the backlog one at a time along with the corresponding data. To improve overall quality and accelerate the work, we use extreme programming practices and techniques and especially programming in pairs: two developers, both from CGI or one each from CGI and the client, work at the same computer to replatform one application at a time.





Factors	WF	Target Application
1	Codebase	
2	Dependencies	
3	Config	
4	Backing services	
5	Build, release, run	
6	Processes	
7	Port binding	
8	Concurrency	
9	Disposability	
10	Dev/prod parity	
11	Logs	
12	Admin processes	



To do so, we put the application in version control, install CI/CD pipelines, target the release process to the runtime environment, and make as few changes to the application as possible while ensuring that it will run in its new environment. Automating the build, test, and release process lets us do this by trial and error. As soon as the application runs without any errors, the modification and migration process is finished, a release pipeline is in place, and your application can be deployed in the target environment with full automation. It is now cloud-ready. In a further step, although usually only if there's a business need, the cloud maturity of an application can be increased by applying more of the Twelve-Factor App principles to make it more resilient, add more mature APIs and so on. This modernization path leads to a cloud-friendly, cloud-resilient, and ultimately cloud-native state.

After we have migrated a few applications, patterns typically emerge. These are specific to your business and applications. Our engineers record these patterns in an application modernization cookbook that we create during the replatforming project. Later you can use this cookbook and the recipes it contains to involve more of your people.

# The Rebuild Approach

While small- and medium-sized applications can feasibly be rebuilt from scratch, our rebuild modernization approach is very effective for modernizing very large monolithic applications that have evolved over the course of decades and are crucial for the success of your business. A project of this kind requires great attention to detail and a reliable partner who is able to support an ongoing modernization effort and make sure that the required resources are consistently available.





Rebuilding an application this way is a major project that can take many months or even years to complete. We generally start by creating a work structure. Like in our replatforming modernization approach, its key elements include a continual assessment stream and a continual replatforming stream. One major difference here is that we don't have to search for applications that can be minimally modified to run in a native cloud environment. Instead, all of the involved people (business, IT, management etc.) are guided through the process of defining business domains, bounded contexts, entities, value objects and so forth. We apply strategic-domain-driven design at the program level to identify the priorities and requirements for the next six months.

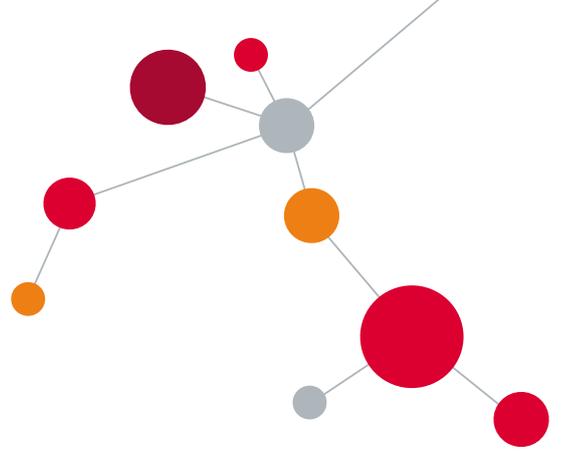
Next, we hold event storming workshops to apply tactical domain-driven design techniques together with IT and business people. This generates enough detailed information to populate an agile backlog with epics and user stories. Then multiple teams start rebuilding (mimicking) the functionality of the existing legacy application as a cloud-native micro-service running in a cloud-native environment. This service is then integrated with the existing application via APIs (the functionality is rebuilt outside of the legacy application). We use all of the same cloud-native application development best practices for the rebuilding work as in our replatforming approach. Everything is automated with evolved CI/CD pipelines, and the resulting newly created software is cloud-native based on Twelve-Factor App principles. Service by service, the existing application's functionality is gutted. This process is repeated until either all of the functionality has been moved out and rebuilt or there is no business motivation for rebuilding any leftover functionality in the old legacy application.

# Modernization on a Global Scale

It isn't easy to carry out high-quality replatforming and rebuilding modernization work. And it's even harder to do so on a global scale while delivering consistently high quality at an attractive price. The described replatforming and rebuilding approaches are CGI products that we are continuously improving and evolving. All of our IT modernization units – in the United States, Europe, India and elsewhere – take the same approach and share the same dedication to delivering quality work to clients all over the world.

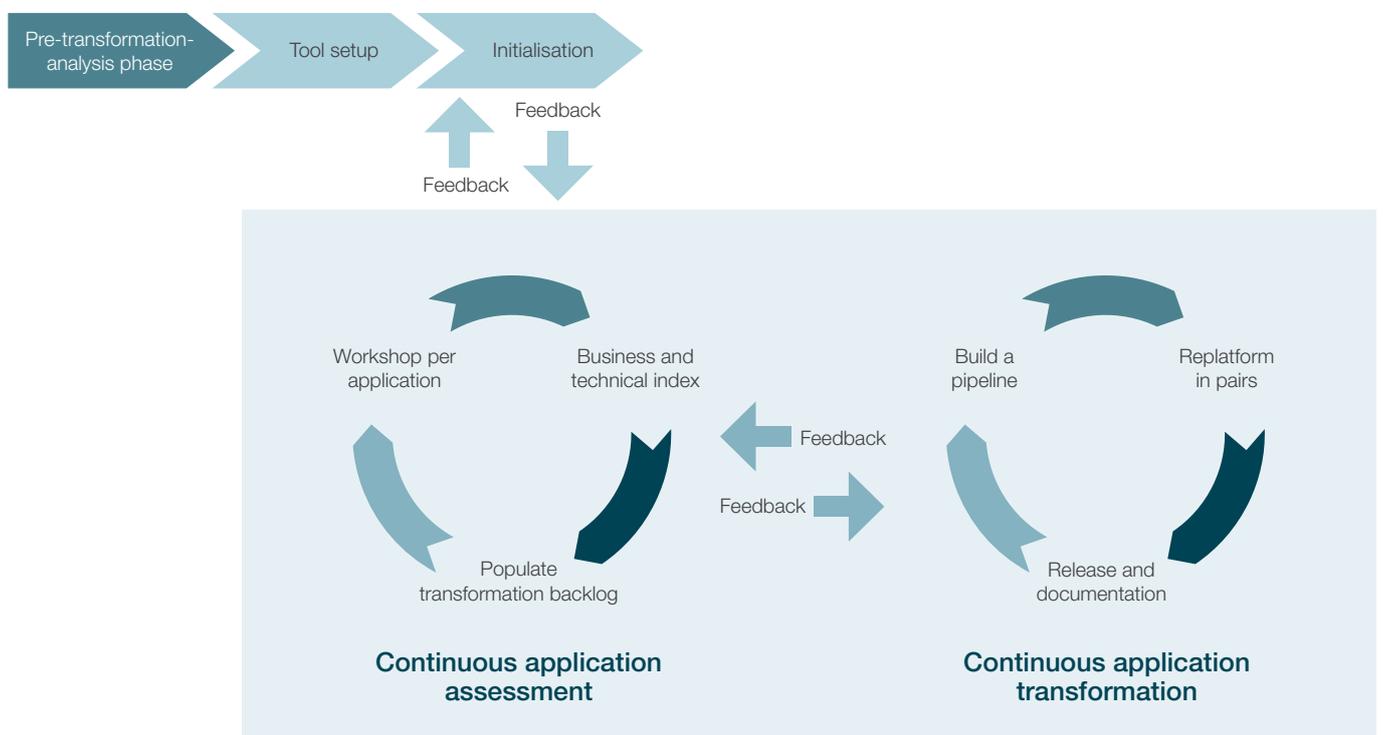
Our unique IT modernization approach, our deep familiarity with the “old” and “new” IT worlds, our client focus, our commitment to excellence, and our ongoing efforts to cultivate long-lasting relationships are all good reasons for you to consider CGI for your projects to migrate to cloud-native solutions.





# Getting started

As Eric Ries recommends in his book *The Lean Startup*, “think big, start small, scale fast”. We wholeheartedly share this view.



Starting small gives you a chance to understand better how applications are transformed with our IT modernization approach and how it meets your needs. In an initial assessment, we identify potential candidates for modernization, which are usually small to mid-sized applications about five to seven years old. Quickly transforming a few applications can generate a momentum that, if supported by the right measures, can grow, increase overall performance and deliver huge benefits. If you're interested in learning more and discussing your company's path to cloud-native applications, contact us now!

# Conclusions

Many companies across most industries are now experiencing a need to modernize their applications. There are many reasons for this. Sometimes the biggest driver is the requirement to reduce infrastructure costs. Or the need to get products and services to market faster and achieve greater flexibility in the development process. In many cases, all of the above applies. With our IT modernization services, we do more than help you migrate your applications to cloud environments. We also apply industry-specific know-how and best practices gained from project work around the globe to find the best approach for your specific situation. Optional programming in pairs by our and your experts adds more value by injecting know-how into your organization. Whether your challenge is to migrate hundreds or thousands of existing legacy applications to cloud-based infrastructure or to rebuild highly complex large legacy systems, thanks to our engineering skills, application transformation approaches, global delivery capabilities, and client proximity model we're able to smoothly guide you through even the most complex application transformation projects.

We're looking forward to hearing from you!





**CGI Deutschland B.V. & Co. KG**

Leinfelder Straße 60  
70771 Leinfelden-Echterdingen  
T.: +49 711 72846-0  
E-Mail: [info.de@cgi.com](mailto:info.de@cgi.com)  
[www.de.cgi.com](http://www.de.cgi.com)