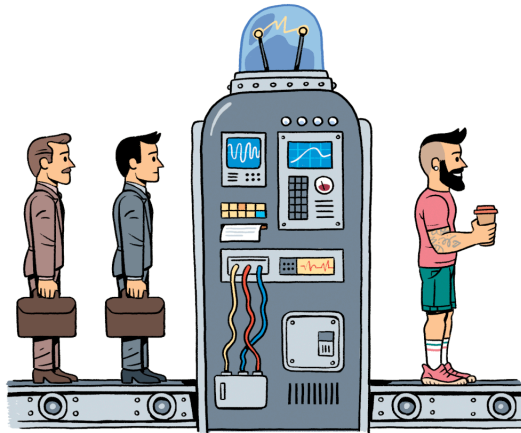


# 8. Retain/Re-Skill/Replace/Retire

The four “R”s of migrating your workforce.



WMAF = Workforce Migration Acceleration Framework

When IT leaders and architects speak about cloud migrations, they usually refer to migrating applications to the cloud. To plan such a move, several frameworks with [varying numbers of R’s](#) exist. However, an equally large or perhaps even bigger challenge is how to migrate your workforce to embrace cloud and work efficiently in the new environment. Interestingly, there’s another set of “R’s” for that.

## Migrating the Workforce

New technology requires new skills. Not just new technology skills like formulating Helm charts, but also new mindsets that debunk existing beliefs as described in “Reverse Engineering the Organization” in *37 Things*. For example,

if an organization equates “fast” with “poor quality”—they do it “quick and dirty”—they will consciously or unconsciously resist higher software delivery velocity. Therefore, *change management* is a major part of any cloud journey. Organizations are made up of people, so people are a good starting point for this aspect of your cloud journey.

## The 4 “Rs” of People Transformation

When it comes to preparing your workforce for the cloud journey, just like with applications, you have several options:

### Retain

Although cloud computing permeates many aspects of the organization, not all jobs will fundamentally change. For example, you might be able to keep existing project managers, assuming that they haven’t degenerated into pure budget administrators. Retaining staff might also include providing additional training; for example, on Agile methods or a shift from projects to products.

Interestingly, even though [applications built for the cloud are different](#), developers often fall into the *Retain* category because they are already used to adopting new technology. Chances are they have already tinkered with cloud technology before your organization went “all in”. The most important aspect for developers is that they don’t see cloud computing just as a new set of tools but understand the fundamental shifts, such as [making everything disposable](#).

### Re-Skill

Other roles might remain by title but require completely different skill sets. This is often the case in infrastructure and operations for which processes that were once manual are now automated. This doesn’t mean folks will be out of work (after all, [automation isn’t about efficiency](#)), but the nature of the work is changing substantially. Instead of (semi-)manually provisioning servers or other IT assets, tasks now focus on building automation, baking golden

virtual machine images, and creating more transparency through monitoring.

*Re-skill* should be the first choice in any organizational transformation because, as I like to say, “the best hire is the person you didn’t lose.” Existing staff is familiar with the organization and the business domain, something a new person might still have to acquire over several months. Also, the hot job market for IT talent has hurt the signal-to-noise ratio for recruiting, meaning you have to sift through more candidates to find good ones.

## Replace

Not every person can be re-skilled. In some cases, the jump is simply too big, whereas in others the person isn’t willing to start from scratch. Although it might be easy to peg this on the individual, for folks who have excelled in a particular task for several decades, having the intellectual rug pulled out from under their feet can be traumatizing.

Its worth considering that being replaced isn’t always bad in the long run. So-called “out of date” skill sets often end up being in high demand due to dwindling supply. For example, many COBOL programmers make a very good living!

Naturally, replacing also implies the organization finding and onboarding a new person with the desired skill set, which isn’t always easy. More on that in a moment.

## Retire

Some roles are no longer needed and hence don’t need to be replaced. For example, many tasks associated with data-center facilities, such as planning, procurement, and installation of hardware are likely not going to be required in a cloud operating model. Folks concerned with design and planning might instead evaluate and procure cloud providers’ services, but this function largely disappears after an organization completes the move to the cloud. In many IT organizations, such tasks were outsourced to begin with, softening the impact on internal staff.

In real life, just like on the technical side of cloud migration, you will have a mix of “migration paths” for different groups of people. You’ll be able to retain and re-skill the majority of staff and might have to replace or retire a few. Choosing will very much depend on your organization and the individuals. However, having a defined and labeled list of options will help you make a meaningful selection.

## You Already Have the People

Categorizing people is a whole different game than categorizing applications—this isn’t about shedding the “legacy”! When looking for new talent your first major mistake might be to look only outside the organization. You’re bound to have folks who showed only mediocre performance but who were simply bored and would do a stellar job once your organization adjusts its operating model.

In a [famous interview](#), Adrian Cockcroft, now vice president of cloud architecture strategy at AWS, responded to a CIO lamenting that his organization doesn’t have the same skills as Netflix with, “Well, we hired them from you!” He went on to explain that, “What we found, over and over again, was that people were producing a fraction of what they could produce at most companies.” So, have a close look to see whether your people are being held back by the system.



In my talks and workshops I routinely show a picture of a long row of Ferraris stuck in a traffic jam (or perhaps a cruising event) and ask the audience how fast these cars can go. Most people correctly estimate them to have a top speed north of 300 km/h. However, how fast are they going? Maybe 30. Now, the critical question is would putting more horsepower in the cars make them go faster? The answer is easy: no. Sadly, that’s what many organizations are attempting to do by claiming they “need better people”.

You might also encounter the inverse: folks who were very good at their existing task but are unwilling to embrace a new way of working. It’s similar to the warning label on financial instruments: Past performance is no guarantee of future results.

## Training Is More Than Teaching

One technique that can be a better indicator than past performance is training. Besides teaching new techniques, a well-run training that challenges students as opposed to just having them copy and paste code snippets (many of us have sadly seen those) can easily spot who latches on to new tools and concepts and who doesn't. I therefore recommend debriefing with the instructor after any training event to get input into staffing decisions. Training staff also gives everyone a fair shot, ensuring that you don't have a blind spot for already existing talent.

Training can also come in the form of working side by side with an expert. This can be an external or a well-qualified internal person. Just like before, you'd want to use this feedback channel to know who your best candidates are.

Some organizations feel that having to (re-)train their people is a major investment that puts them at a disadvantage compared to their "digital" competitors who seem to employ only people who know the cloud, embrace DevOps, and never lose sight of business value. However, this is a classic "the grass is greener" fallacy: The large tech leaders invest an enormous amount into training and enablement, ranging from onboarding bootcamps over weekly tech talks and brown bags, to self-paced learning modules. They are happy to make this investment because they know the ROI is high.

## Top Athletes Don't Compete in the Swamp

Many organizations are looking to bring in some super stars to handle the transition. Naturally, they are looking for top athletes, the best in their field. The problem is, as I described in a [blog post](#), that the friction that's likely to exist in the organization equates to everyone one wading knee-deep in the mud. In such an environment a superstar sprinter won't be moving a whole lot faster than the rest of the folks. Instead, they will grow frustrated and soon leave.



On my team we often reminded ourselves that for each task we have two goals: first, accomplish the task, but also to improve the way it's done in the future.

So, you need to find a way to dry up the mud before you can utilize top talent. However, once again you're caught in a chicken-or-egg situation: draining the swamp also requires talent. You might be fortunate enough to find an athlete who is willing to also help remove friction. Such a person might see changing the running surface as an even more interesting challenge than just running.

More likely, you'll need to fix some of the worst friction points before you go out to hire. You might want to start with those that are likely to be faced first by new hires to at least soften the blow. Common friction points early in the onboarding process include inadequate IT equipment such as laptops that don't allow software installs and take 15 minutes to boot due to device management (aka "corporate spyware"). Unresponsive or inflexible IT support is another common complaint.



I was once forced to take a docking station along with my laptop because all equipment comes in a single, indivisible bundle. When returning equipment I ran into issues because some of the pieces I didn't want in the first place were tucked away in some corner.

HR onboarding processes also run the risk of turning off athletes before they even enter the race. One famous figure in Agile software development was once dropped off the recruiting process by a major internet company because he told them to just "Google his name" instead of him submitting a resume. I guess they weren't ready for him.

## Organizational Anti-Corruption Layer

You can't boil the ocean, so you also don't need to dry up all the mud at once. For example, you can obtain an exception for your team to use non-standard hardware. However, you must dam off the part that's dry from the rest of the swamp lest it floods again.

I call this approach the *Organizational Anti-Corruption Layer*, borrowed from the [common design pattern](#) for working with legacy systems. In organizations, this pattern can for example be applied to headcount allocation:



When working in a large organization, I had all team members officially report to me so that shifting resources between subteams would not affect the official org chart and spared us from lengthy approval processes.

You can build similar isolation layers for example for budget allocation.

## Up Your Assets

One constant concern among organizations looking to attract desirable technical skills is their low (perceived) employer attractiveness. How would a traditional insurance company/bank/manufacturing business be able to compete for talent against Google, Amazon, and Facebook?

The good news is that, although many don't like to hear it, likely you don't need the same kind of engineer that the FAANGs (Facebook, Apple, Amazon, Netflix, Google) employ. That doesn't imply that your engineers are not as smart—they simply don't need to build websites that scale to one billion users or develop globally distributed transactional databases. They need to be able to code applications that delight business users and transform the insurance/banking/manufacturing industry. They'll have tremendous tools at their fingertips, thanks to the cloud, and hence can focus on delivering value to your business.



In a past job, I was quite successful in hiring qualified candidates because our team didn't require them to speak the local language (German). Additionally, many of the “digital brand-name” competitors had very few actual engineering positions in their sales and consulting offices, which also required extensive travel.

Second, large and successful organizations typically have a lot more to offer to candidates than they believe. At a large and traditional insurance company, we conducted an exercise to identify what we can offer to desirable technical candidates and were surprised how long the list became: from simple items like a steady work week and paid overtime (not every software engineer prefers to sleep under their desk), we could offer challenging projects on a global scale, access to and coaching from top executives, assignments abroad, co-presenting

at conferences with senior technologists, and much more. It's likely that your organization also has a lot more to offer than it realizes, so it's a worthwhile exercise to make such a list.

## Re-Label?

In many organizations, I have observed a “fifth R”: the re-labeling of roles. For example, all project managers underwent SCRUM training and were swiftly renamed into SCRUM masters or product owners. Alternatively, existing teams are re-labeled as “tribes” without giving them more end-to-end autonomy. With existing structures and incentives remaining as they are, such maneuvers accomplish close to nothing besides spending time and money. Organizations don't transform by sticking new labels on existing structures. So, don't fall into the trap of the fifth R.