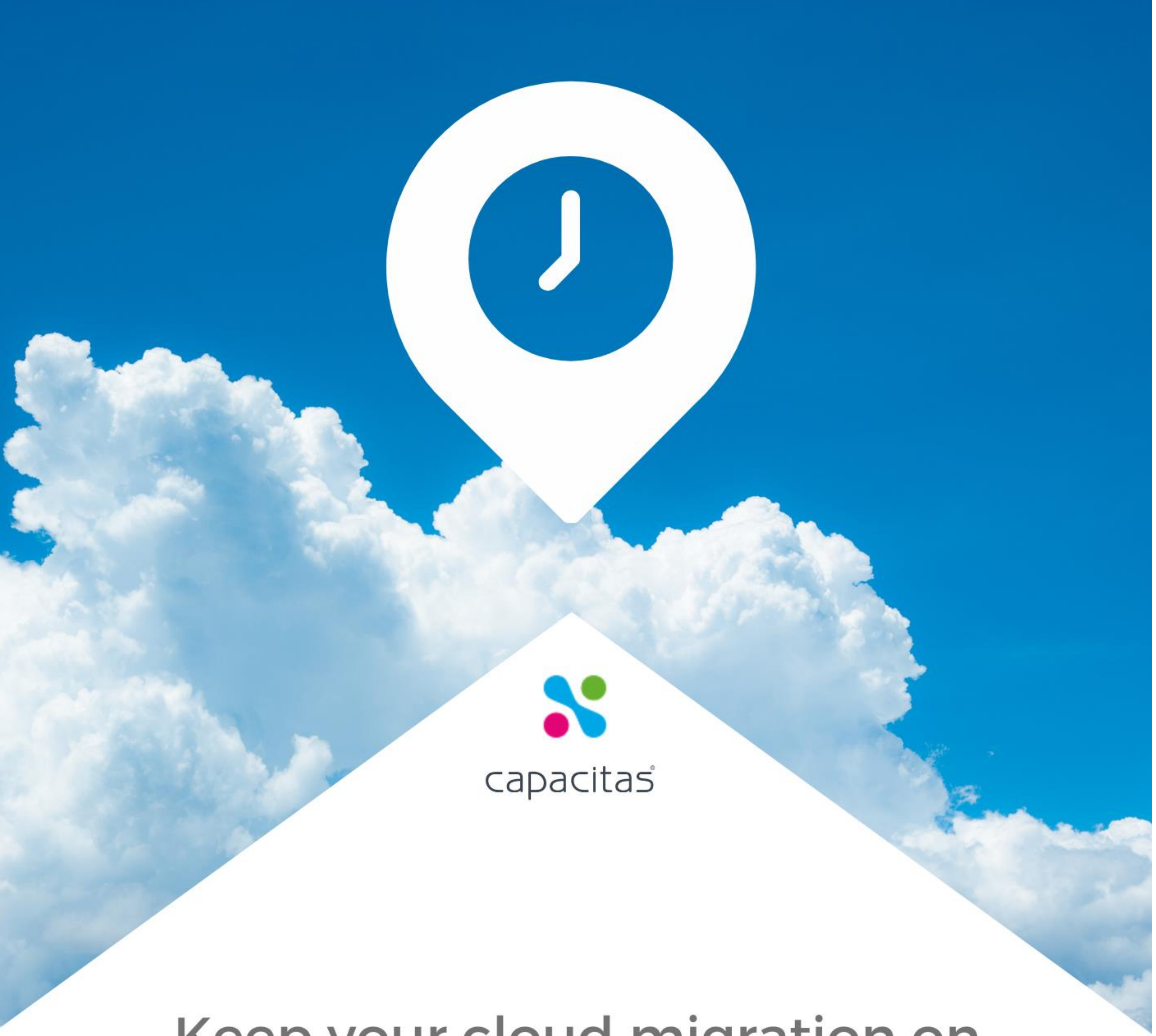

Keep your cloud migration on time and on budget following 3 steps

Dr. Manzoor Mohammed

There is enormous pressure from the board to migrate to the cloud from existing datacentres. The timescales are very tight. This is because the migration must be complete before the existing datacentre contract expires. Any contract extension incurs severe financial penalties. These can be into millions of dollars.

There are other financial penalties of delay. Delays in migration mean that RI/savings plans and/or associated credits may be wasted. Teams can also consume credits ahead of budget, because they have not adapted cloud working practices. If that wasn't enough, there is also the pressure of keeping the existing services up and protecting revenue.

Delays and cost overruns in major cloud migration projects are typically due to 1) Unexpected technical constraints 2) Legacy working practices and 3) Applications not optimised for existing environments. (1) (2)

Keep your cloud migration on time and budget following these 3 steps:

1. Lift and shift in stages – act fast with real data

2. Model and Fix at each rollout stage

3. Remove waste as you go

1. Lift and shift in stages – act fast with real data

The sooner you get real users then the sooner you understand how your system performs and costs, lift and shift is a good approach as you will get real data on how change impact user experience. Real data is always better than data observed from test systems.

The most important thing is determining performance and cost of the system as you roll out. There are 5 ways of doing this:

1. **Analyse key metrics for the end-to-end IT system:** Define expected key business, service and resource metrics. These could be business metrics e.g. conversion rate, service, user response time and resource metrics, CPU, memory utilisation. You need to agree reporting granularity and time windows, e.g. 1 minute over 2-week time window.

2. **Use a common reporting tool set:** As well as agreeing metrics you need to agree where and how it is going to be measured. There will be differences if one user is using Omniture and another is using Google analytics. This will cause confusion and delay the project roll out.

3. **Have clearly defined stages during roll out:** A well planned roll out schedule will allow you to look at these key metrics without ambiguity. It’s easier comparing metrics and seeing if they are better or worse. One way of doing that is to have easy analysis stages, e.g. 25%, 50%, 75% and 100% workload migration. See diagram below.

4. **Model the expected system sizes for each stage:** Avoid going to full sizes systems when you only have partial roll out. Full size systems not only burn up budget but will mask underlying technical constraints. It’s better to have an appropriately sized system for

each stage. This maintains user experience but does not mask early warning signs of technical constraints. This will also give you an early warning sign of unexpected cost.

5. **Model the expected cost for each stage:** Have a model of what each stage of the roll out will cost. This should be broken down by service (e.g. webserver, application server etc) and capacity type (e.g. compute, storage, disk IO etc).

2. Model and Fix at each rollout stage

Use the data to get early warning signs to identify technical constraints or cost issues at each migration stage. These could impact your project timelines and budgets. If you have a warning of either performance or cost not being in line with the model, you can put together a plan to fix this and bring it back in line by the next stage of the roll out.

There are 6 steps to doing this:

1. Validate size and cost models at each stage. Where there is an anomaly investigate why and where the difference exists. From your preparation work you will have broken down costs by service and resource type. This will allow you to narrow down where your original assumptions were wrong.

2. Model what future system performance will look like at each stage of the roll out. You need to go in deep and look at the 13-key metrics of each component. Doing a deep dive will give early warning signs of unexpected issues. These could be I/O, network constraints or costs that you didn't expect when you modelled the system.

3. Don't leave the most active users till last. One customer we dealt with left their most active users till the end of the migration. Unfortunately, these users were the ones which caused the most problems. This caused a large delay to the project timelines. Ideally, you want to migrate a portion of these active users in the first stage to understand the risk of these users at full roll out.

4. Re-model future cost profile based on observations at each stage. At the first stage you should have a sign if your costs are going to be within budget. If you don't then you will use your model to track where the difference is and what that means for costs of subsequent phases. This will also inform you on whether you want

to do any optimisation prior to the next stage.

5. Optimise as you migrate where there is a risk to either timelines or budgets based on your modelling insights. If you are seeing issues in the early stages then you will need to optimise. This could be simple change or refactoring of code. Having the access to the right metrics and expertise will allow you to make the right decision.

   Most companies have APM tools (e.g. new relic, appdynamics etc.) and resource monitoring (e.g. datadog, Zabbix etc). These tools make the collection of these metrics easier. Also, there may be opportunities to assess whether you need to maintain performance levels and unnecessary processing.

6. Be confident that you can size downwards as well as upwards. The cloud instances are likely to be more powerful than your previous boxes. This is especially true for applications that use lots of CPU. Anything you save while maintaining performance can be used for future budget risks.

3. Remove waste as you go

Engineers will need to start learning how to do remove unnecessary capacity at every stage of the migration activities. Getting embedded this into the culture early on will start upskilling your engineers to build more cost-effective systems and also manage their costs better for in the future.

There are 7 typical waste removal activities that you need to do to keep your migration within budget:

1. Avoid duplication of systems and data. Engineers set up test systems to provide assurance that systems will work in the cloud. This can lead to duplication of data or unnecessary data volumes.

2. Avoid oversized test systems. It may be tempting for the engineers to use the largest instances possible when testing. The ideal mindset is wanting the smallest test system possible to do the job. This is especially true in functional environments.

3. Remove Idle instances. There will be always be instances that left on. That's normal we are human and it’s easy to forget to turn off an instance. Set up scripts to remove idle instances especially in test environments. Some companies have automatic schedules that turn off systems overnight and on weekends.

4. Limit oversized production systems. Engineers will be tempted to give their applications the best performance possible. You can't blame them, they want to give their users the best experience. The reality not all applications deserve to be treated as gold.

5. Make cloud costs visible to engineering teams. Reports showing individual team costs and breakdown are useful. It also helps start culture of housekeeping. It needs to be supported with people’s time to explain what the reports mean and where there could be areas of

improvement.

6. Define and implement a tagging strategy – this is important otherwise you won’t be able to allocate costs.

7. Remove migration specific systems after project is complete. It’s common to have systems that were in place to support the migration to remain even on after migration is complete.

The earlier these methods are adopted, the easier the migration will be

It may seem like a distraction to look at this now. Analysing and modelling all this data at each roll out stage seems time consuming and complicated. Who will do it? Your engineers will be focused the cloud migration. But, with the right resources these activities can done in parallel with your engineers.

Time spent in these three areas will make sure your project stays on track. Once you have a working model it will become business as usual. You will understand your users, applications and the cloud in a way that you never could before. This understanding you allow you manage cloud costs effectively as part of BAU. It will also allow you to invest in areas to improve user experience and efficiency.

Having a plan that allows you to see early warning signs is critical. Good modelling methodologies and working practices supports this. This gives actionable insights early on to tell you where to invest your time to keep the project on time and within budget.

References

1. (n.d.). Retrieved from https://www.cpomagazine.com/cyber-security/lift-and-shift-cloud-migration-benefits-disadvantages-and-use-cases/

2. (n.d.). Retrieved from https://www.jeffersonfrank.com/aws-blog/top-reasons-cloud-migrations-fail/