

# The Seven Sins of Multi Cloud

Kris Saxton, Partner & Co-Founder



**Automation Logic**  
Better business through automation

## 1. Hybrid-Cloud vs. Multi-Cloud

There is often confusion around what multi-cloud is and what it is not which can lead to unmet expectations. Multi-cloud is a new term coined because Hybrid Cloud was getting overloaded with concepts like on/off prem, brokers and abstractions layers as well as just running more than one cloud provider in parallel (which is what multi-cloud is).

Multi-cloud means adopting and running workloads on more than one cloud provider at the same time. They may be managed by common tooling and process, but they are fundamentally separate and workloads cannot be moved between providers at run time.

## 2. Software vs. Traditional IT

Multi-cloud platforms, like all modern infrastructure, are strange beasts. They have aspects in common with the software services they support and aspects of the infrastructure services they are born from yet are leaving behind.

Software is eating the world, as Marc Andreessen said, and infrastructure is no different. Failure to recognise this means the benefits of implementing and managing a service in software vs traditional infrastructure approach means we'll fail to recognise the benefits in terms of speed, quality and cost.

By traditional IT we mean older software tools that don't recognise the primacy of the network connected API.

The impact of this is that engineers are given inadequate tools and so the levels of automation they can achieve are stymied. And automation frequently only delivers the benefits when the entire process is automated from engineer to end-user so it only takes one bad apple in the pipeline to ruin the party for everyone.

### 3. Trying to be an IaaS

One of the enormous benefits of multi-cloud is the move away from shared services to codified patterns of cloud use. This is where you have governed access to IaaS and PaaS services (expressed as code) but without actually creating any persistent infrastructure that has to be fed and watered.

The key benefit of this approach is very small code base and ops teams and therefore much lower running costs. But ops teams tend to create persistent infrastructure by default. Why? Because it's what they're used to and they don't really see the penalties with having things to manage (because it's them managing them and that's what they see as their job).

To manage this, organisations need to embed strong architecture principles that actively encourage the avoidance of persistent infrastructure and strong product ownership within infrastructure to implement that vision and get teams focused on supporting devs rather than building things they'll have to manage.

### 4. Building too much upfront

One of the main reasons for frustration with delayed delivery can be traced to this sin, whereby a decision is made to ensure that anything built is fully compliant with an organisation's policies and processes from day one. And as a result it appears as if nothing of value is being delivered for a long time, because end users and even some stakeholders don't care about these compliance measures, they aren't seeing anything they could use.

Strong product ownership should keep the team focused on building 'potentially shippable units of value' and showing those regularly to stakeholders. This is the best way of keeping everyone engaged (and so funding) work that often takes much longer than regular software projects to actually produce something that can be used in production.

## 5. Overcomplicating

Engineers are curious and are driven to build things; it's ingrained so deep, it's part of their DNA. But without strong guidance they will build a ladder to the moon, just to see if they can and because it's just so shiny.

Platforms built by engineers will generally be awesome. Awesome technically but also awesome in terms of their cost and the size and skill of the teams required to maintain and run them. If building platforms with governance and security in mind produces delivery goals that are too broad, then engineer led delivery's tend to be too deep. Again, strong product ownership with a laser focus on doing as little as possible to meet the user need are the key here.

## 6. "Faux Agile"

Drawing on the recent writing of Martin Fowler, who described the initial challenges of the agile community as relating to getting people to take agile seriously and do it at all. Now the problem we see is programmes being labelled as agile in name, but what is being practiced is far from it. "Faux Agile" can be serious killer of both morale and delivery velocity.

What's needed is much bigger and ongoing investment in IT leadership, so they aren't just doing agile for the sake of it, but they have the deep and lasting understanding that will survive the pressure of a deadline.

## 7. Org transformation

Beyond technology, organisations need to be prepared to manage the people and process implication of cloud. What we're seeing is that, whilst the low level delivery teams have successfully reinvented themselves around the concept of the product or the service, the surrounding ecosystem through which these deliveries are usually coordinated and supported remains largely unchanged.

Some of this can be attributed to the way IT programmes are funded and but also a general failure in leadership to grasp just how far you have to go to take cloud adoption deep into the enterprise. Everything you have to do to make agile and digital stick, goes for cloud adoption too. Most people we speak to outside of engineering, don't think or see how this affects them. It's business as usual from their perspective.

The lessons learned in engineering about the fundamentals of lean and agile, are just as powerful and relevant to people working in finance, HR and programme management, and that's where we have to invest, that's where we have to go next, bringing everyone with us.



## The Virtues of Multi-Cloud

### Abstraction

Make sure you know the difference between multi and hybrid and why you're adopting multi-cloud.

### Ambition

The Product Owner role has no equivalent in the traditional IT programme and so the role is frequently forgotten completely or given to the lead engineer or delivery manager. Recognise its importance and resource it properly through specific hiring or training.

### Approach

Leaders are letting their programme leadership down by sending them into battle flying the Agile flag without adequately arming them with the knowledge for the task at hand which leaves everybody unsatisfied. We've done it for engineers, we need to spread lean and agile thinking into other parts of the org, starting with the programme and it won't happen naturally. You have to invest in bringing everybody with you.

---

### Want to know more about better business through automation?

Speak to Emily Toole on +44 (0)7484 661 395 or email [emily@automationlogic.com](mailto:emily@automationlogic.com).





**Automation Logic**  
Better business through automation

T: +44 (0)20 7183 4204  
E: [info@automationlogic.com](mailto:info@automationlogic.com)  
**[automationlogic.com](http://automationlogic.com)**

17 Wootton Street  
London SE1 8TG  
United Kingdom