SKELTON THATCHER
EFFECTIVE SOFTWARE OPERATIONS

Whitepaper: changes to technology & teams needed for modern software systems

Skelton Thatcher Consulting Ltd, January 2015

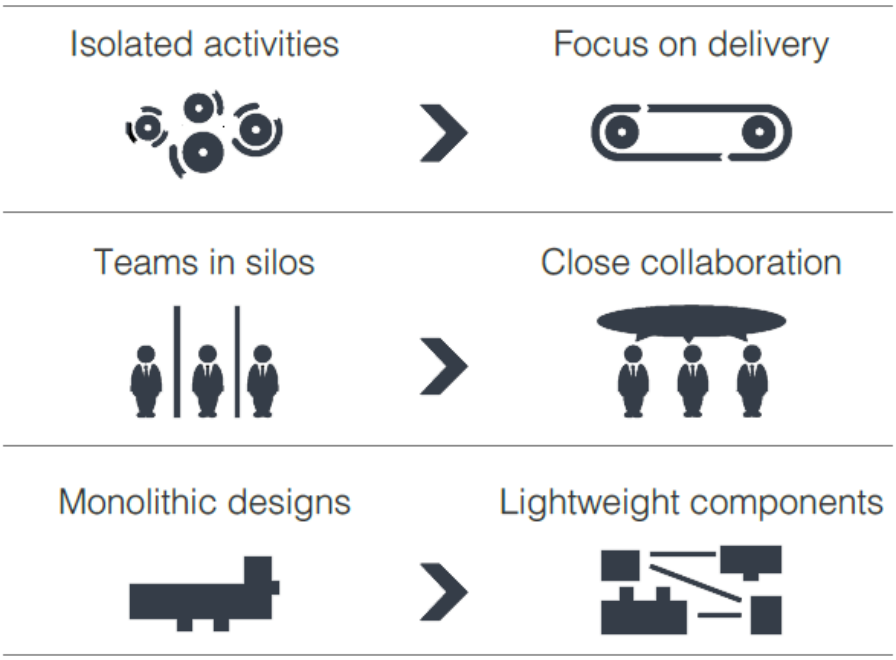
'Cloud' changes everything for organisations building and operating their own internet-connected software systems. However, many organisations are unaware of the scale and nature of the changes in technology and teams needed to take advantage of cloud in order to remain current and competitive.

Faster delivery comes at a price worth paying

The pre-cloud model of software delivery assumes that the order and speed of new features has little impact on the longer-term viability of the software system. This 'fire and forget' approach is not suitable for ever-evolving cloud-based software systems. Instead, the continuous innovation practiced by successful organisations includes all teams in the value stream, including IT operations and service desk staff.

Techniques and practices such as Continuous Delivery and DevOps work well precisely because 'hand-over' between teams is avoided in favour of in-team responsibility for both innovation and improvements based on feedback from the software in operation. Such approaches may not be cheaper in terms of head count, training, or salaries, but the resulting approach – seeing IT as 'value-add' – can be much more effective than one that sees IT purely as a cost centre, as recent studies by Gartner, Forrester, and others have shown.

Lean manufacturing has shown that by limiting work in progress (WIP) for teams, organisations can achieve *more* in a given length of time, but this requires discipline from the commercial or product teams and a clear strategy for product evolution. Cross-product prioritisation with limited WIP is well-suited to cloud software.



Elastic, programmable, on-demand infrastructure – 'cloud' – was pioneered in the late 2000s by Amazon with Amazon Web Services (AWS) and since taken up by numerous additional cloud providers. The lead time for IT infrastructure changes in the cloud (whether on-premise or hosted) can be mere minutes or seconds. This step change in the time taken for infrastructure changes – along with the greater control, visibility, and arguably security afforded by programmable 'infrastructure as code' – has huge ramifications for organisations building and operating differentiated software systems and services.

t: +44 (0)20 8242 4103 | e: enquiries@skeltonthatcher.com | w: skeltonthatcher.com

Operability should be a first-class concern

Cloud requires us to *continue to care* well beyond the initial deployment into Production. This means that both user-visible and operational features must be prioritised from the same backlog and by the same product owner, who must also be measured on the both user-visible features *and* the operational success of the system. In parallel, people in an operational role must be considered as stakeholders with requirements by the software development team.

Two key operational criteria essential for cloud software are *deployability* and *monitorability*. The capability to deploy new software in a rapid, reliable, repeatable, and recurring manner is absolutely fundamental to a successful cloud-based software system. Allowing deployability to shape the design of software is no more or less strange than allowing deployability to shape the design of temporary bridges in a battlefield situation; the changing temporal requirements drive the architecture.

Likewise, the need for monitoring is paramount in cloud-based systems, partly because we simply cannot predict all behaviours when dealing with an open-ended complex distributed system. Cloud software needs dedicated monitoring features such as diagnostics endpoints along with auxiliary tooling for metrics collection; these cannot be retrofitted afterwards, but must inform the system design.

Team structures dictate systems architecture

One of the most astonishing realisations about software systems – first articulated by Mel Conway in 1968 – is that an organisation building software is constrained to produce designs that reflect the communication structures of the teams within the organisation: Conway's Law. This has huge ramifications for organisations building differentiating software systems. It suggests that large, up-front designs by software architects are doomed to failure unless the designs align with the way in which the teams communicate.

In turn, it means that by restructuring teams, and facilitating (or potentially deliberately *limiting*) communication between teams, we have a much better chance of building systems that work well in Production and feel natural to evolve over time. That is, if we know we need to be able to deploy different parts of the system independently with a short Lead Time, and we decide to use small, decoupled services in order to do so, we need to make our teams similarly small and decoupled, with clear boundaries of responsibility.

Summary

In order to take advantage of the many benefits of cloud technologies, organisations that develop and operate their own software systems must set up their organisations in way radically different from in the past. Team structures must match the required software architecture or risk producing unintended designs. Software must be designed with operability as a first-class concern, with the ability to deploy and monitor the systems being core to the software's features, otherwise endangering the long-term viability of the software. Finally, the way in which the development and operation of the software systems is funded must encourage collaboration and treat IT as adding value rather than incurring cost.

Skelton Thatcher Consulting can help you transform your technology and teams for the cloud.
Call us on +44 (0)20 8242 4103 or email us at enquiries@skeltonthatcher.com for more details.