

Reshape Your Legacy

5 burning issues that mean Legacy Modernisation should be a priority

Many organisations rely on an assortment of core legacy systems to keep their day-to-day business running and to support their Unique Selling Proposition (USP). However, they are facing rising consumer expectations and increasing competition from a growing number of new, agile start-ups who are harnessing disruptive technologies to bring innovative products and services to market faster. This dependence on an ageing infrastructure is becoming ever more expensive and problematic across all areas of a business – not just for those working in IT, but also departments such as Finance and Procurement. We look at five ways legacy portfolios are negatively affecting organisations, and discuss why this needs to be urgently addressed.

1. Legacy skills shortage

A recent survey from Forrester Consulting found that 57 per cent of enterprises with a mainframe currently run over half of their business-critical applications on the platform; and this number is set to rise to 64 per cent by 2019. However, businesses are struggling to replace a dwindling pool of skilled mainframe workers. Organisations have lost 23 per cent of these employees in the last five years and 63 per cent of these positions have not been filled.*¹

In a 2012 Computerworld survey, 46 per cent of respondents said they were already noticing a COBOL programmer shortage, while 50 per cent said the average age of their COBOL staff was 45 or older and 22 per cent said the age was 55 or older*². That was six years ago. Which means those COBOL programmers are now aged between 51 and 61, or older, and they are retiring now.

Similarly, IBM Systems Magazine found that 85 per cent of respondents to their 2016 survey agreed that a mainframe skills gap exists. In the same survey respondents stated that 18 per cent of their mainframe staff will be retiring by 2021, taking many years of experience and vital knowledge with them.*³

It's not just COBOL programmers. Legacy platform operations and support skills are all in short supply. When you consider other legacy platforms, such as OpenVMS or Fujitsu VME for example, the resource issues are even more dire. It is increasingly difficult to find any employees to keep these environments running.

92 out of the 100 top banks in the world, and 23 out of the top 25 US retailers, use IBM mainframes. There has been talk for many years about the impending retirement wave of legacy skilled Baby Boomers and in that time there have been calls for training

Reshape Your Legacy

younger programmers in mainframe and other legacy skills. However, this hasn't happened. Universities and colleges are not adding COBOL to their computer science syllabuses, and Fortran and Assembler have become the equivalent of Latin or Sanskrit – ancient languages that need a 'Professor' to interpret them. And yet, there are many critical applications in production today using not only COBOL, but PL/1, Fortran and even Assembler.

The industry has seen the problem coming, done nothing to mitigate it and now we are at the point where legacy skilled Baby Boomers are actually retiring and leaving the workforce. Not only does this create resourcing problems to maintain, enhance and run legacy systems, it also creates a brain drain of decades-worth of knowledge exiting IT departments.

2. Interoperability with legacy systems creating more complexity

Digital Transformation strategies are very often focused on developing new systems of engagement, providing new mobile, Cloud-based applications for a company's customers, business partners and employees. Digital Transformation is also increasingly about IOT (Internet of Things), which means an explosion of Cloud-connected devices streaming data back to an organisation. And yet, very often, all of these new digital applications and devices are required to connect back to legacy, mainframe-based systems of record.

This need for interoperability between new Cloud platforms and old legacy platforms has created a requirement for middleware to connect the two together. In general, there are products available on the market to connect newer apps to older ones, or integrate real-time data streams along with data from transaction processing databases. Over time, in order to support new digital initiatives, organisations have often had to implement a variety of middleware solutions. These range from message queues, enterprise service buses, and a wide variety of Web Services interfaces – frequently using proprietary solutions for enabling legacy applications to consume web services and be consumed as web services.

In some cases, however, there are no easy and standard ways to integrate into legacy applications and data. This leaves companies with legacy 'islands of computing' that cannot be leveraged as part of a seamless processing system with newer applications, except by using out-of-date file transfer methods.

Whether you have various middleware layers to enable legacy interoperability or you are struggling with the problem of not being able to integrate into your legacy systems, this issue at a minimum creates complexity and additional cost in your IT landscape. At its worst lack of interoperability becomes a real blocker to Digital Transformation initiatives.

3. Technical Debt

Technical Debt is defined as a concept in software development that reflects the implied cost of additional rework caused by choosing an easy solution now instead of using a better approach that would take longer. In other words, it's all the quick and sloppy coding that has gone into an application to get a project finished on time. Technical Debt can be calculated as the time/cost it would take to redevelop those areas of rushed coding into compliant, elegant, maintainable code.

Although coding is the main area of focus, Technical Debt can also include lack of technical documentation for an application and poor code coverage for testing. So, the areas that are typically considered as Technical Debt are:

- > Poor source code formatting, not adhering to standards
- > Lack of modularity
- > Code complexity
- > Low test coverage
- > Lack of documentation

When looking specifically at legacy applications we can add another category of Technical Debt to the list. Because some legacy systems are decades old they frequently contain coding that was done in a specific way 10, 15 or 20 years ago to deliver the required functionality. However, if you were to develop that functionality today you wouldn't do it that way. Technology and programming techniques have moved on, but companies are left maintaining old, out-of-

Reshape Your Legacy

date coding to support specific functionality. This legacy Technical Debt not only carries a potential cost of redevelopment with it, but also an on-going operational cost because the coding can be so complex and so old that programmers stay aware from it and don't want to change it. The code becomes black box, 'no-go zones'. This slows down enhancements and significantly impacts an IT department's ability to respond to the needs of the business and requirements of their customers.

So, Technical Debt has both an implied cost of remediation, but for legacy applications also a real cost of ownership.

4. Software Entropy

One definition of entropy is lack of order or predictability; a gradual decline into disorder. For applications, Software Entropy can be summed up by two key factors:

- > A computer program that is used will be modified
- > and, when a program is modified, its complexity will increase, provided that one does not actively work against this.

What does this mean for legacy applications? Well it's not good. Many core legacy applications have been around for 10 or maybe 20 years, and in some cases even longer than that. They have all been modified - over and over again - and the modifications have been done by a variety of programmers. Some of who cared greatly about avoiding adding complexity and disorder to the application code, but also some who really didn't care about those issues at all.

There is no doubt that for the majority of legacy systems, over time as modifications and enhancements have been applied, complexity has gone up. However, the increase of complexity is not even across the entire application. Frequently, over time, applications develop highly complex outliers. This small number of core processing programs are critical to the application's functionality, but have become so complex they are almost impossible to understand - and any change to them usually comes with the potential to break them. In some cases, these programs get to the point where they are 'no-go zones' and instead of modifying the program, developers avoid touching them and instead design additional layers around them to provide the enhancements required.

In general, it is fair to say that the longer an application has been in use, the more modifications it has undergone. As a result, it increasingly becomes more difficult to enhance and more likely that any changes will cause unknown errors or problems elsewhere in the application. Over time, legacy applications become 'brittle'.

One reason for legacy Software Entropy is that the majority of legacy applications were not developed in Object Oriented programming languages. This means that most legacy applications are less modular in their design compared to modern applications, and do not have good mechanisms for code reuse. 3GL languages, like COBOL, over time tend to create large monolithic programs with numerous logic paths through them.

As organisations look to modernise legacy applications, approaches that can preserve the business logic in an application, but move it to an Object Oriented structure, will help reorganise the application to avoid complexity 'hot spots' and reduce potential Software Entropy going forward.

5. Legacy - the high cost of ownership

There are plenty of statistics suggesting that maintaining and operating legacy applications consumes anywhere from 60 per cent to 80 per cent of corporate IT budgets. The previous four points in this paper explains why this may be. The cumulative impact of the issues created by maintaining a large legacy portfolio is an increasing cost of ownership. As legacy resources become more and more scarce, then the cost of these resources goes up further. The need for layers of interoperability software to bridge newer applications with the older legacy systems, creates additional costs that wouldn't be required if the applications were a more homogeneous technical architecture. Technical Debt is a measure of cost in its own right. Remediating old code may not be a cost you wish to incur, especially if it is not affecting day-to-day operations, but not incurring this cost will likely have the effect of increasing Software Entropy. This in turn will increase the time and cost of development projects.

Reshape Your Legacy

Added to the cost impact of the factors we have discussed, are the additional licensing and operational costs of legacy software and hardware. Many mainframe applications make use of old databases and utility software that is disproportionately expensive when compared to modern counterparts. Mainframe and other legacy hardware platforms also carry considerable operating costs when compared to commodity computer platforms or especially the Cloud.

The combined effect of all this is that legacy applications and infrastructure create a cost drag on an enterprise, diverting valuable funds that could otherwise be directed to digital innovation that would bring competitive advantage.

Case in point

Balfour Beatty Utility Solutions, one of the UK's premier contractors, operates in multiple markets including energy, rail, public and industrial sectors. Alan King, their Chief Engineer, commented, "Our legacy mainframe required a major upgrade, and there was obviously the ongoing and increasing maintenance costs, and inherent risk of running our business-critical product design suite on a proprietary platform."

Balfour Beatty Utility Solutions migrated to an open and more flexible Windows environment to ensure they continue to enjoy the benefits their system provides, and an increasing return on their investment – all without the need for user retraining. Alan King sums up, "With our migration to Windows we were able to deploy this critical application on a far more cost-effective platform, while ensuring we could retain the competitive advantage gained from this bespoke

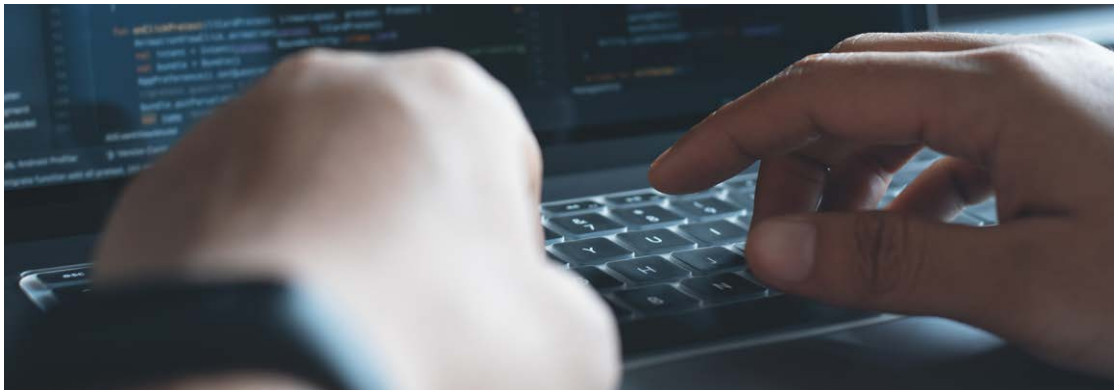
system." They can continue to run and enhance their system in a modern environment, removing reliance on the dwindling pool of programmers with legacy skills and creating greater interoperability with modern systems.

Legacy Modernisation – removing the barriers to Digital Transformation

A recent Forrester report on Digital Transformation recommended that organisations, "be bold and move quickly – most technology executives have been too conservative changing out back-end systems." According to Forrester, "Real Digital Transformation requires fast, sweeping modernisation of old systems into new Cloud-based systems that are agile, mobile and able to deliver disruptive, real-time intelligence."⁴

Every organisation today is primarily a technology company, no matter what product or service it provides. Before digitalisation there was a clear demarcation between tech companies and non-tech companies. The switch from analogue to digital blurred the lines as organisations struggled to stay relevant and started to deploy technology solutions to reimagine their business. Today, the most successful and highest performing businesses have little separation between their business strategy and their technology strategy.

Reshape Your Legacy



With technology at the heart of a company's operations, much more than a business enabler, it becomes business critical. So critical in fact, that if the technology infrastructure failed it could, in theory, bring the organisation to its knees. This is why it is becoming increasingly dangerous to ignore the need to modernise core, critical legacy systems.

Modernisation projects can be complex and disruptive. They require specialised skills, methodologies and tools. However, they need to be urgently addressed as any barriers preventing these critical systems from being agile and responsive to business needs are damaging. With the right approach, these modernised legacy systems can be the foundation for an organisation to thrive in the digital era, building on many years of specialised knowledge within different departments, while enabling the digital innovation and responsiveness that customers now expect.

***1** <https://www.cbronline.com/news/cios-facing-die-mainframe-moment>

***2** <https://www.computerworld.com/article/2504568/data-center/the-cobol-brain-drain.html>

***3** <http://ibmsystemsmag.com/mainframe/trends/whatsnew/skills-survey/>

***4** Forrester: "Your Digital Transformation Is Not Bold Enough – Five Signs Of Trouble And Key Fixes" - May 2017

More information

w oneadvanced.com
t +44(0) 8451 605 555
e hello@oneadvanced.com

Ditton Park, Riding Court Road, Datchet, SL3 9LL

Advanced Computer Software Group Limited is a company registered in England and Wales under company number 05965280, whose registered office is Ditton Park, Riding Court Road, Datchet, SL3 9LL. A full list of its trading subsidiaries is available at www.oneadvanced.com/legal-privacy.