devoteam

2020

5 problems caused by outdated software

Creative tech for Better Change

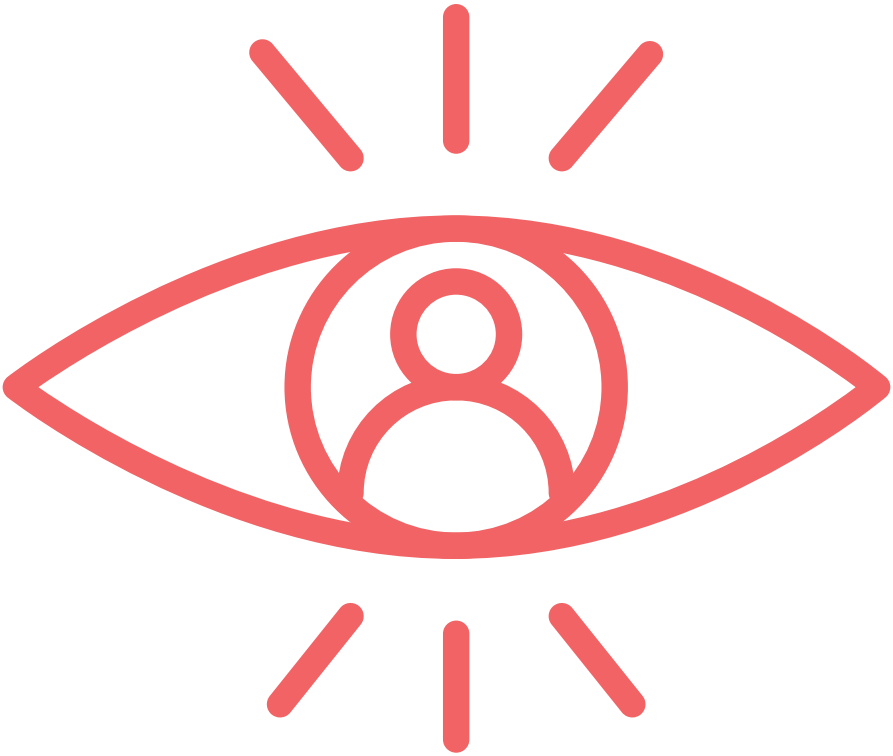
TABLE OF CONTENTS

INTRODUCTION

This is the first article in our series on Software Modernization. Every piece of business software is a product of the time and environment it is created in. More so in older methodologies that see the maintenance phase of the Software Development Life Cycle (SDLC) as the end of development than in agile methodologies, the capacity for software to radically break with the central tenets of its development is often limited by this fact.

Even when software is maintained perfectly, and some features have been added over time to meet some previously unknown requirements, there comes a time when the world around said software has simply changed so much that it starts to get outdated. It is no longer able to be adapted quickly to new requirements, as they are too different from what was required at its inception.

When this happens, the outdated software can have a significant impact on the business as a whole, and can lead to a host of problems. In this article, we explore some different business problems that can be caused by outdated IT systems. Of course, there are many more problems to be identified, however we have selected 5 to serve as examples of the trouble outdated software can cause.



PROBLEM 1: INCREASED MAINTENANCE COSTS

The first effect of outdated software we want to highlight is perhaps the most obvious one; cost. There are several ways in which outdated software can lead to higher maintenance and operational costs. This is an example of the problems we categorize as operational challenges; problems which arise while simply maintaining functionality as-is.

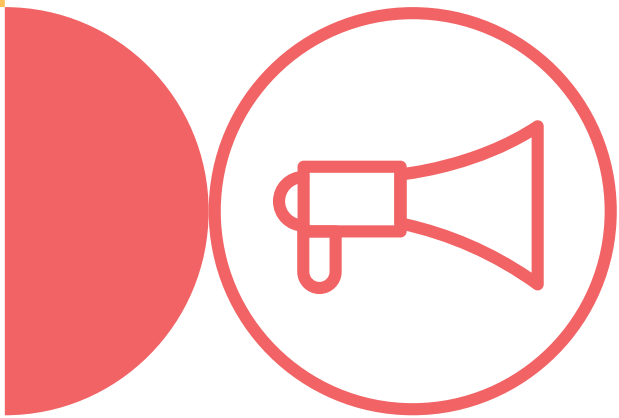
A) Some software requires specific hardware

Some software has been written to be executed on specific hardware. This can be anything from a Point-Of-Sale device to a mainframe. Issues arise when, after a certain time, this hardware fails and needs to be replaced. Finding spares for older equipment can be a difficult and costly process, and code tailored specifically for certain devices limits the possibilities of transplanting the software unto more modern hardware or cloud environment

B) Finding personnel capable of working with outdated software stacks can be a challenge

Software stacks like programming languages and semi-customized off-the-shelf systems require specific expertise to create functioning, performing, secure software. The availability of engineers with the right skillset to support a chosen software stack is therefore of vital importance to a companies' ability to maintain software written on that stack.

Trends in software development change rapidly, so the current generation of engineers might not be as able to work with an older software stack as the previous generation was. Finding engineers capable of working with an older software stack can get to be a challenge after some time, as governments and large companies still running COBOL software are finding out. As the pool of skilled engineers decreases, rates increase.



C) Code complexity tends to grow over time

Most business software is never finished. As time goes on, feature changes or additions, bug fixes and changes in coding style can increase the complexity of a system. After years of changes, the increased complexity of code can steepen the learning curve for new developers, and can make executing changes take longer. Increased complexity also increases the chances of bugs being introduced in new changes, and can lengthen the testing process, leading to increased time-to-market.

D) Knowledge will have gotten lost

One often overlooked cause of increased maintenance costs in older software is the loss of knowledge about that system. At the inception of a piece of software, some people within the company tend to know exactly what the software does, both from a business perspective and a technical one. Those people know all the quirks and features, and the reasoning behind their existence. Unless an extensive effort is made to convert all this implicit knowledge to explicit knowledge, and the documentation is continuously updated, knowledge will get lost.

The architect who designed the software might leave the company, or the developers who know every part of the code by heart. Documentation might have gotten lost. Losing knowledge about the working of software increases the time it costs to resolve bugs, and to build new features.

PROBLEM 2: INABILITY TO INNOVATE

(LACK OF FLEXIBILITY)

The category of problems we have dubbed technical limitations, revolve around the inability of outdated software to meet the changing requirements that come with the evolution of a company. When a new piece of software is created, it is bound by the paradigms and technologies of its time. Though some expansion on both technical and business sides might be anticipated, the rate of societal and technological changes in the last decades has been such that planning for the future is a near impossible task.

Companies might want to expand into new markets, or add products to their portfolio. If a warehousing system cannot cope with new types of products, this might seriously hinder the capacity of the company to respond to changing market demands.

A) Covid-19

The recent COVID-19 outbreak and related economic measures have highlighted the importance of being able to have staff work from home. Older software lacking networking capabilities can be a serious problem for companies affected by these measures. With no possibility to have staff work from the office, a company that cannot keep their employees working from home is in serious trouble. Schools faced with an immediate need to facilitate studying at home, can be seriously hampered in their educational effectiveness if they cannot rapidly integrate video conferencing systems into their application landscape.

B) Online Retail

Currently compounded by the effects of COVID-19, which causes more people to order online instead of visiting physical shops, e-commerce has steadily grown over the last years. In 2019 alone, US e-commerce sales grew by 16.6%, while physical sales only grew by 3.9%. E-commerce sales in the Netherlands grew by 21% in the same year.

From vendors of digital goods like games and films to supermarkets, being able to sell products online has become a necessity for nearly every company. For example, current inventory numbers need to be made available by API, and automatic ordering has to be made possible. Without the ability to sell online, an ever greater source of revenue will go untapped, and the company's image might suffer as competitors are perceived as more modern.

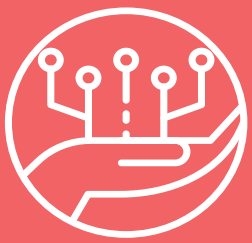
C) Data & Insights

Data fuels 21st century business. Extracting the right information from the company allows executives to make the best decisions, allows the company to find and fix problems faster, gives insights into customer behaviour, and helps optimize business processes. While modern IT systems are often designed with the need for efficient data-gathering and monitoring in mind, older systems can predate this mindset. IT-systems that do not allow monitoring and data-gathering can deprive the company of information needed to make the best business decisions.

D) Data & Insights

With data being called the 21st century equivalent to oil, being able to connect different data sources and to trade data, has become critical to some companies' business success. With data platform providers like Google and Facebook leading the way, more and more companies create a layer of API's over their systems enabling rapid integration with business partners, customer systems, suppliers and government systems. Creating a set of easy to use API's can encourage those external parties to connect with your systems, and share their data in return.

Not being able to supply an API landscape, or to consume API's published by business partners, can hinder the streamlining and optimization of business processes. Older software, created without the requirement for data integration in mind, can impede this ability.



PROBLEM 3: INABILITY TO SCALE

(INCOMPATIBILITY WITH CLOUD)

Being able to rapidly scale up, down, out and in, allows a company to react to peak loads on IT systems caused by, amongst others, holidays, marketing campaigns and unforeseen events. Missing holiday season sales because of overloaded IT systems or being overwhelmed by traffic due to a very successful marketing campaign can be costly both in financial sense and in NPS. The most popular way of enabling rapid scaling is migrating to a public cloud like Amazon's AWS, Google's GCP or Microsoft Azure. Older software can be difficult or impossible to move to a cloud environment because of very specific platform requirements, restricting scaling ability.

PROBLEM 4: SHADOW IT

When older software does not completely meet requirements anymore, but still functions good enough to not alarm business, complaints from people working with the outdated software might be dismissed.

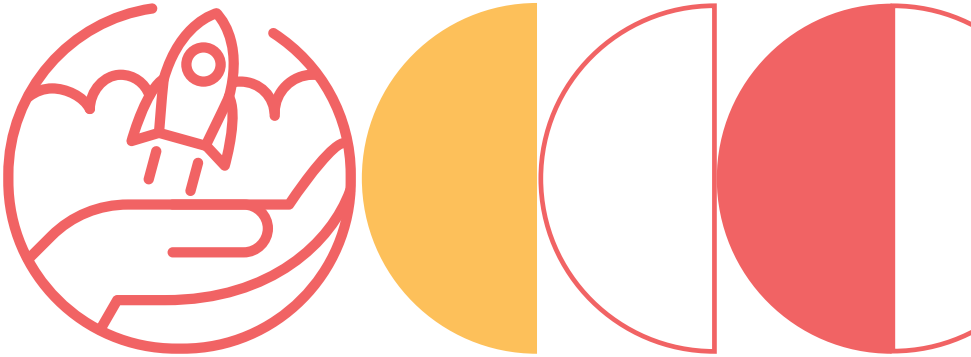
This can cause people to think of their own ways to compensate for the perceived lack of software capabilities. A landscape of unofficial tools and processes can grow, comprised of Excel sheets, macros, tiny undocumented tools written by well-meaning hobbyist developers, and unsafe usage of the outdated system.

This poses a business risk, as such unofficial tools are often not well documented or tested, not under version control, or only usable by their creator. If, for example, an Excel sheet that compensates for a financial system that cannot cope with new regulations gets lost or contains bugs, the company might suffer financially.

PROBLEM 5: LEGAL & SECURITY RISKS

For companies operating in the EU, the introduction of the General Data Protection Regulation of GDPR has meant sweeping changes to the way personal information should be handled and stored. Breach of the GDPR regulations can result in heavy fines being imposed on the company responsible for the breach. Software that cannot be updated to deal with the requirements imposed by GDPR can be a serious legal risk for a company.

System and data security too have rapidly gotten more important, as the number of hacks and data leaks have increased. A data leak can seriously damage a companies' image, and, next to legal ramifications, can lead to serious drops in revenue and NPS. Hacks and leaks of confidential company information like blueprints and legal documents also poses serious business risks. Older software can be more susceptible to attacks and hacks if it is not patched or updated regularly, or has not been written with current security considerations in mind.



CONCLUSION: COMPETITIVE DISADVANTAGE

The list of 5 problems described in this article is not an exhaustive one; in different situations, there are many more problems that can be caused by ageing IT systems. We have selected these 5 problems based upon the frequency of occurrence we have noticed.

What these problems have in common is that they can impact a company both in a direct and an indirect way. Direct effects can be felt and measured by the company itself, and concern missed revenues, high costs, and increasing risks.

The indirect effect of these problems on a company is that they cause competitive disadvantages compared to competitors who have a more modern IT-landscape which allows them to cut costs, tap more revenue sources, and respond to changing market conditions quicker.

NOW IS THE TIME TO MODERNIZE

The first step of any software modernization program is finding out why you are considering it. What are the reasons software modernization sounds like an appealing proposition for your company? A clear view of the reasons for starting this process can help in determining what actions to take, and how to measure success. Want to learn more about Software Modernization? Have a look at our white paper Software Modernization: 7 considerations before starting.

At Devoteam we have a strong Software Engineering practice and our roots in the areas of DevOps and Integration. With this combined skill set we are able to guide companies in their digital battles. Our +/-300 technology consultants guide clients through their digital transformations from day to day, from integration, API Management, Cloud, and Microservices to DevOps, Automation (CICD), Software Engineering, and Data Integration. Working together with the world's leading technologies such as Google Cloud, Red Hat, Microsoft, and Mulesoft, Devoteam functions as a consulting and implementation partner for clients such as Liberty Global, Vodafone Ziggo, Eneco, Rabobank, various municipalities/government institutions, and many more.

CONTACT

Jasper Baljeu
Author / Software Engineer
jasper.baljeu@devoteam.com
Devoteam

Melis Schaap
Director Business Development
melis.schaap@devoteam.com
Devoteam

devoteam

#TechforPeople