Original Article

Extensible software-as-a-service: Stability and agility

Jordan Gray

obtained his application development experience since 1999, starting with Revolutions, Inc., a medical equipment dotcom called Spinlife.com. This dotcom experienced incredible growth with Jordan filling multiple roles of CTO, Web Strategist, Creative Director and Web Marketing Director. Today, he is the Inventor-in-Chief at Quick Square Consulting, which he co-founded with Matthew Yerkes, who is the ideal counterpart. Friends since childhood, together they have assembled the right team to invent, build and sustain expertly engineered business applications. You will especially enjoy talking with Jordan if your company is trying to solve a high-stakes problem that involves Web applications. He is intensely curious about discovering the complex interrelationships and dynamics that exist in business between people, things and ideas. Jordan has a B.S. from The Ohio State University. More info at: <http://www.linkedin.com/in/jordangray/>.

ABSTRACT It's important to have a 30 000 foot view discussion about some of the software options available to a company, particularly when they are seeking a DAM solution. With DAM being at the core of most all business industries and segments, choosing the wrong technological path could be devastating. This easy going discussion about the landscape of software options discusses fundamental strengths and weaknesses of each. Recognizing limitations of certain types of software puts companies in the best position to choose a solution that will be the most sustainable and least restrictive for years to come. If stable but flexible technologies are not utilized in business, the company will be burdened by the technology instead of being helped by it. Learn the allures and pitfalls of canned versus custom software. Understand the exponential impact of a failed technology path. Recognize hidden cost of ownership and opportunity costs in software. Learn how software can fail before it is even deployed. Learn how SaaS, if extensible, could be the most sustainable software model yet.

Journal of Digital Asset Management (2010) 6, 323–326. doi:10.1057/dam.2010.45

Keywords: extensible software-as-a-service; xSaaS; hosted; web based; digital asset management

INTRODUCTION

Harsh realities are challenging most of what we believe about sustainable governance in many areas of our lives – software technology not excepted. Although businesses that touch digital images and media are adapting to new trends in media, their software struggles to stay relevant. Teams of workers either find workarounds for software limitations or are hamstrung by their own technology.

Business owners often have to stop everything and fix gaping technology holes in their work flow or business process. Sometimes they invest countless hours investigating *new* out-of-the-box software solutions hoping it will

last longer than the previous solution.

Unfortunately, canned software solutions indeed have a shelf life.

If a company doesn't know how to cook up IT solutions, it ends up buying a canned product. Chances are the next meal again will be off the shelf. As sure as the next quarter in business, changes will arise that force kluge fixes by the team or shopping for more canned goods.

TECHNICALLY GIFTED PARADOX

Paradoxically, some the most technologically gifted companies may be in the most trouble – because they take the bait. They cook up their

Correspondence:

Jordan Gray
Quick Square Consulting,
3510 Snouffer Road,
Ste 200, Columbus, Ohio
43235, USA

own solution. Compromising focus on their core business, they dedicate valuable team resources and funds to create a *custom fit*. A tailored solution seems within reach. So, they sneer at ill-fitting canned solutions and go to the other extreme on the continuum. But the answer is somewhere in the middle.

In tech savvy companies, technocrats within the organization hear the issues and develop a solution to be the perfect fit. If the goods are delivered as promised, they call it a win. But just as they are packing their bags and slapping their high-fives, a few more project requirements suddenly appear. The bane of custom software solutions is *feature creep*. These new requests of the software arise partly from failures in communication, discovery, planning, scoping and partly from new emerging realities in business. Half of the problem is precisely the same as the failures with canned solutions, except the custom solution may end up costing more than had been bargained for.

LIMITATIONS IN SOFTWARE VETTING METHODS

Some non-technologically gifted companies fall into the trap of shopping the existing solutions based on *perceived value*. This is an oversimplified approach to choosing a technology solution as it does not examine current business requirements, possible trends related to workflow and current and future access needs. (Taking this approach could allow a company to peek into the future, estimating and gauging needs that arise even in the short term.) Instead, those seeking software solutions believe that they will pick a winner if they chart out a nice big grid of comparison points and run the numbers. The one with the most features for the least price wins.

Caveat: A solution's limitations are seldom on the list. Limitations of a solution are the most expensive part of any quote. Without an understanding of limitations, the cost of ownership cannot be calculated. (Dead end straight ahead.) And the only way to know the limitations is to fully vet the business requirements and debate potential future trends that may emerge. Once those things are quantified, they can be measured against the features and flexibility of a solution. This is the only way to discover likely limitations and the costs involved of those limitations.

SOMEONE NEEDS TO CHOOSE

Suppose you are the one chosen to select technology needs for your company. On the one hand, companies are drawn to tried-and-true – the predictability of canned software and stable solutions. However, your business needs, media types and output options keep changing. (If only the business world would stop moving, you might have a sporting chance to figure out what it needs.) You realize that all of these changes will put a stressor on custom solutions just as much as canned solutions.

The hunt for the right software solution can lead your team down many rabbit trails while consuming valuable time and company resources. How do you avoid such pitfalls when you're on the hunt?

Instead of being sucked into either extreme, check the middle ground to see whether there are some options. Interestingly, Software-as-a-Service (SaaS) has proven to be a viable middle ground for many companies. (It's outside the scope of this document to address all the pros and cons of SaaS.) As a whole, it has the potential to enjoy better stability than custom applications. As its community of users' business needs evolve, it quietly provides timely updates.

These updates are typically the most critical improvements and features needed by the SaaS community at large. But it is unlikely that your business will follow the same path of the community – even if the SaaS is targeted to a particular industry. Few companies are alike. So there is no promise, no guarantee that SaaS by itself will be perpetually relevant to your evolving workflows or new business requirements. SaaS offers a possible middle ground which we will further explore later in our discussion. But businesses today remains ambivalent. On one hand it wants the cheapest chicken, and on the other it wants a perpetually custom tailored solution. Their heart's desire? Business should be leading the dance, instead of technology, but it also cannot break the bank.

ALLURES OF CANNED AND CUSTOM

Let's take a look at what is great about both of these extremes. Is there a way to get the best of both worlds? If you consider what the attraction is for the two software extremes, and know their pitfalls, you may be able to identify a better model for building sustainable software solutions.

Allure #1 – one-time expense

Although it's true that canned software typically offers the most immediate, budget-fitting, short-term benefit in *out-of-pocket* savings – ticket price is not everything. These canned products can be sold cheaply like any other manufactured, mass-produced item. This software is designed with the most amount of features to work for a broad audience. Often, there are so many features that it is not uncommon for many of them to be irrelevant – to overload the user and decrease adoptability. (We are not going to discuss how this kind of 'most affordable' software also forces workers into disruptive work patterns trying to work around inefficiencies.) In addition, one-time costs rarely do not include opportunity costs. These can only be fully quantified by the business development director or CEO. They are the ones who know what business potential was lost while technologically hamstrung.

Allure #2 – custom fit

The allure of custom-fit software for a specific business is often perceived as the biggest win of all. With the possibility of a proprietary system comes competitive advantage. The eyes get wide thinking about having control of every software feature. But is it really achievable or even desirable? What are the real costs? It turns out that building custom software is not much different than commissioning craftsmen and artisans to produce a work of art. The process is often fraught with communication failures, assumptions and dashed hopes.

FAILURES IN THE EXTREME

Sadly both of these extreme software options are very prone to fail – victimized in proportion to the dynamic nature of a given industry. Full featured *canned solutions* take a shotgun approach to solving as many business needs as possible. On the other hand, *custom* software aims to satisfy *current needs* – but only as much as they are properly defined or anticipated.

Just for kicks, let's imagine a perfect world where custom software reflects precise business requirements as defined and communicated. Further, the custom build-out happens as expected and delivered on schedule. The project's long-term success would still be affected by the emergence of new business issues.

Feature creep is usually something that programmers and application developers try to protect against, often with a view that the request for new features is slightly malicious having been announced 'after the fact'. However this problem is two-fold. First, the application developer did not, or was blocked from completely vetting the project requirements presented. Reality is that building custom software is like trying to jump a moving hurdle. (And it's your 'teammates' who appear to be moving the hurdle.)

Changing requirements is not any one person's fault, rather a product of a dynamic system and communication failures. This drama is not just for the custom software folks. Canned software also deploys with drama. *Software failure starts happening when just one feature doesn't work the way it needs to.*

SaaS ON CENTER STAGE

For sake of the argument today, SaaS enjoys the center stage position, bringing with it stability, lower costs and more rapid updates that are most relevant to the user base. It's the SaaS community at large that takes the software through its paces. Subscribers to the software have a powerful voice. It's a voice requesting new features, reporting problems – these issues are not ignored. *SaaS keeps its subscribers and its subscribers keep SaaS.* These SaaS members are pay-as-you-go and vote with their feet (if needed, they'll walk away). Thus, SaaS communities of subscribers enjoy relevant updates and new features that are widely beneficial to the members. Some might say that SaaS is just a next generation canned solutions, but overall it does canned better.

The problems with canned software can still repeat themselves inside of SaaS if the community grows very large. In such cases, the community voting power is weakened and voice is lost. For example, when a large SaaS community launches major upgrades, it can be overbearing and unwelcome. Think of the impact on users in a SaaS receiving updates and new features that appear out of thin air. A crude example of this backfire with users is after an upgrade inside of Facebook – a large community SaaS unpaid by its users. A group formed in Facebook after a new version was deployed. Today there are over 1.6 million

users in this group called ‘We Hate The New Facebook, so STOP CHANGING IT!!!.’ So yes, features and upgrades can happen automatically in SaaS, but the new features may not be needed or appreciated.

The biggest problem with plain-old-SaaS occurs when its business subscribers can’t wait to get a critical feature, but the community doesn’t realize the same need. You could wait a digital lifetime and your SaaS community may never add that golden feature. So again companies can find themselves hamstrung without the flexibility of having a critical business requirement met by the software. Great hopes for the software to be a technological holy grail – again dashed. Companies have to then resign themselves to using me-too technology instead of differentiating in their industry.

At this point, technology is placing a burden on business, when it should be providing a mechanical advantage. Companies are lulled into existing with me-too-software solutions which makes the business more easily replicated. (Danger: Global market ahead.) When trying to win new customers, and all of your competitors are using the same solution as you? Customers will not be able to tell the difference.

ADDING A CUSTOM LAYER TO SaaS

Alas, there is a way out of being stuck with me-too software. Extensible Software-as-a-Service (xSaaS) allows for companies to apply a proprietary unique layer to their SaaS. This provides all the benefits of SaaS (incremental costs, increase stability) and adds features that are specific to your industry and emerging business goals. It’s alive! It’s a unique skin that allows a thin layer of custom features instead of an entirely custom application. This smaller custom

portion sitting on top of the SaaS itself, is a variant that can be sustained along with regular updates. The reason why this may be one of the most sustainable approaches to software is that it allows for companies to gain all the flexibility and control of custom software at a fractional cost (since it is combined with the cost effectiveness of SaaS). In a way it has brought the best of both worlds together.

STABILITY AND AGILITY

Technology should be subservient to business – not the other way around. xSaaS looks into the face of change and welcomes it – because it can. xSaaS is stability and agility. It’s what we all desire from software. And since business is on the move every year, quarter, month, week and day – we need to ask ourselves the hardest questions about what we believe to be true in the world of software. When we think about past decisions on software, we can be honest and recognize that we made some decisions that were self-limiting. They appeared to be good for the company, but in the end created restrictions. Woody Hayes, the legendary football coach at The Ohio State University once said, ‘You win with people.’ Companies that are the most alive and aware of changes in the market are driven by people who can be agile and respond to new stimuli. But if technology is weighing them down, they cannot perform. People become stronger when technologies are not barriers to their workflows. Identifying barriers to workflow and making nuance changes over time removes technological burdens as they surface. Only then will you be able to imagine your company having the ability to answer new digital market opportunities at speeds that will make the world around you seem like its standing still, because it is.