Whitepaper

Transform IT with Codeless Configuration

ITSM Codeless Technology Survival Guide

Cherwell™

The role of IT in today's fast-paced, ever-changing world is constantly evolving; technology is the undisputed king. The speed and effectiveness with which companies embrace, adapt and implement new technologies determines their ability to be successful in the competitive business arena.

Historically, the IT department was systematic. It functioned to analyze, implement and provide structured software solutions for its users, but generally outsourced the customization of technology in response to pre-defined business needs. In order to support today's fast-paced change, IT teams have greater responsibility for modifying, updating, integrating and changing software on demand. As such, they increasingly seek to implement non-labor-intensive solutions in place, in order to eliminate costly overhead and accelerate change.

This whitepaper will consider how "codeless" software solutions are favorable to those that require customization with a focus on future-proofing, reduced administration overhead and increased flexibility.

The Challenge

The challenges for IT and business are enormous, partly due to the reliance on skilled programmers to customize and adapt software solutions to meet business needs. In 2012, in its analysis of a survey it conducted, InformationWeek noted that "one way businesses are coping with the challenge of upgrading and optimizing enterprise applications is to avoid customizations." In the same survey, InformationWeek found that the barriers to success facing technology professionals include changing, upgrading and customizing internal applications. Of those who responded, 49 percent considered this a major concern. (Henschen 2012)

To understand a problem, it must first be defined, specifically the differences between "customization" and "configuration." Gartner's Brian Iverson provides an in-depth definition of each term, which are often used interchangeably. "Customization is a change that must be performed outside the framework of the product. For example, if I wish to write a rule that reads from information in a table and I must add that table to the product's database using DBA tools, that to me is customization. If the product provides a way to create ad hoc tables for use with rules (and auto-generates the UI for manipulating data in those tables), then the activity would be considered configuration." (Iverson 2015)

Doug Henschen, in InformationWeek, commented on the difference, noting "the difference between customization and configuration boils down to coding and development versus vendor-supplied menu or wizard-driven approach. When you develop custom code to change or extend functionality, there's always a chance the code won't work when the vendor introduces the next release, so extra validation steps are required." (Henschen 2012) Extra steps equal extra expense.

Reliance on a solution that requires "customization" instead of "configuration" is a major ongoing problem. Three primary problem categories influence the issue: maintenance, administration and cost configuration, each an integral piece of the overall puzzle. Let's review them individually to understand the impact.

Maintenance

The maintenance aspect of the issue can be simply stated. In many instances, when software is customized by programmers to change fields, forms, reports and users to meet business needs, the next upgrade will not necessarily support the code and system changes. Customizations will be over-written with default settings and will need to be customized again by costly programmers once the upgrade is complete.

Administration

Administration, specifically integrations with third party and bespoke solutions such as chat tools, Microsoft SCCM, VMware and other solutions designed for a specific business needs are a key element to running a business. However, if an integration is "customized" rather than "configured" not only are you limited by the type of integrations available to you, but when you upgrade the solution, the back-end code is updated, the integration points break, and the latest version becomes incompatible, incurring additional cost and work—a "no win" situation.

Cost

Cost can be prohibitive for most organizations. Teams of programmers with specialized knowledge are required to perform customizations by writing code. Regular process changes performed by programmers to maintain the efficacy of the software become costly and time consuming. Business2Commmunity reported about "black swans," projects that substantially exceeded budgeted cost, leading to not only more bugs in the system but also unwieldly code that needs constant care and attention. (Tomlin 2014)

Clearly, customization is not the solution.

Recommendations

If you want to avoid the pitfalls related to "customizing" your enterprise applications, what should you look for, and where should you look? These are good questions to consider when searching for an IT service management (ITSM) solution.

1. First and foremost, seek a solution that does not "handcuff you to customization" and all of its inherent problems. A codeless solution, one that abstracts administration from the underlying code and database, removes the re-work and extra testing from the equation. A codeless solution should have in depth configuration capabilities allowing you to modify functionality without thinking about or requiring development or coding.

2. The solution should be easy to use. Solutions should be configurable with a GUI/visual builder and drag-and-drop wizards that do not require any coding for any reason – forms, integrations, workflow, reports (no SQL queries) and so on should all be available without touching underlying code.

3. Future-proofing is essential. Any solution will need to be upgraded from time to time in order to take advantage of the latest and greatest functionality. Make these worry-free, and reduce downtime. Codeless configuration ensures your functionality and integrations are maintained from one version to the next.

4. The solution should allow you to share functionality with fellow customers who have similar needs, and the vendor should provide a means for you to connect with those customers. For example, universities often have similar requirements, and one university may configure a time management tool that could be beneficial to another university.

Benefits of a Truly Codeless Solution

This whitepaper discusses the problems that arise when managing and maintaining a customizable solution and how you can overcome these complications by embracing codeless technology. Time savings, agility of change, reduced downtime and simplified integration are benefits that come with configurable solutions.

IT departments can reduce their reliance on costly programmers and turn to more cost-effective resources to maintain, administer and build the functionality that meets their business goals. This, along with the ability to automate manual business processes and maintenance activities, frees IT staff time to focus on higher value business projects that contribute directly to a company's bottom line.

In addition, the ability to respond to and initiate changes in a timely manner will increase flexibility, eliminating the need for more sophisticated change control processes to update your ITSM solution.

Not only can you decrease the cost of solution maintenance, make changes more easily, and extend functionality with a wider range of integrations, but you will have peace of mind that you won't be trapped in an obsolete version due to the likelihood your customizations will break when you upgrade.

Codeless configuration - it is a "win, win" situation.

Change is an inevitable part of the IT business cycle. How a company responds to it is a strong indicator of its potential for success or failure. As long as there are computers, there will be code and systems that are modified, upgraded and integrated into the fabric of the business.

Only one question remains: will you choose technology that requires a clunky, outmoded customization process, with its constant drain on time and resources? Or will you ride the wave of the future with a more agile, "codeless" and therefore configurable solution?

The choice—and the future—are yours.

Henschen, Doug. "Enterprise Applications: Customization Vs. Configuration." InformationWeek. 8 1, 2012 http://www.informationweek.com/software/enterprise-applications/enterprise-applications-customization-vs-configuration/d/d-id/1105616? (accessed 11 29, 2015).

Iverson, Brian. "The Territory Between Configuration and Customization." Gartner. 2 23, 2015. http://blogs.gartner.com/brian-iverson/2015/02/23/the-territory-between-configuration-and-customization/ (accessed 11 29, 2015).

Tomlin, Ian. "Creating Enterprise Apps Without Programming." Business 2 Community. 5 16, 2014. http://www.business2community.com/tech-gadgets/creating-enterprise-apps-without-programming-0885549#ZyCROsQSlsP9krYj.97 (accessed 11 29, 2015).