Configuration Construct

HRM software customization is so yesterday.

By Naomi Bloom

There are two primary methods of ensuring that a specific organization's user experience preferences, work flows, data requirements, processing/business rules etc. are accommodated by HRM software—configuration and, heaven forbid, customization. Before vendors delivered extensive, robust configuration tools (and many still don't!), the usual method of fitting application software to a specific customer was to use delivered exit points/APIs to attach customizing code developed specifically for that customer. And don't even ask about the customizing of delivered vendor code, a common enough practice through the early 90's that has hopefully gone the way of gas guzzling SUVs. Maintenance of this custom code was the customer's responsibility, and it often had to be redone during a traditional on-premises upgrade process. All too often, extensive custom code rendered packages just too expensive to upgrade, and the benefits of using packages was thus consumed by that customization. But doing customizations on a time and materials basis made oodles of profit for systems integrators during their heyday, many of whom have been reborn into BPO teams.

Configuration tailors a package's capabilities to meet a specific customer's needs by using the vendor's delivered configuration tools. We make selections of processing options from pull-down lists, check off desired attributes/formats for reports or inquiries, describe our preferred user experience characteristics from the list of available ones, drag and drop predefined UI components into the desired place on our "home" page, create our own pull-down lists by adding options, build total compensation plans by checking off their characteristics from lego-like collections of attributes and methods, establish our multiple concurrent organizational views by drawing our organization chart(s) graphically, and select the actionable analytics and other features that we believe enhance our ability to achieve important HRM and business outcomes. During configuration, we modify the delivered "good practice" work flows graphically, add new attributes and methods to delivered objects, create entirely new objects along with their relationships, select and modify tables of valid values, and establish who gets to do what within our implementation of the platform. And we don't write any procedural code external to the application—*ever*!

With multi-tenant software, configuration is all that's allowed, so it's particularly important that the vendor has anticipated the full range of configuration capabilities that their customers are likely to need, both initially and over time. One of the early and quite erroneous arguments against multi-tenancy was that it precluded adaptation to meet a specific customer's needs. *Au contraire*. While it's certainly possible to create crummy multi-tenant software without sufficiently powerful configuration capabilities, it's also possible for multi-tenant software to include the best possible configuration tools, thereby creating effective-dated, inheritable configurations across and within tenants. It's also important to note that poor configuration capabilities aren't limited to multi-tenant software, as evidenced by the amount of customization that's still being done to on-premise HRM software.

Great HRM software, multi-tenant or otherwise, enables the widest possible range of such effective-dated configurations to be done, dynamically by properly trained business analysts without relying on programmers to do this work. Obviously, all such changes must be rigorously tested, but the scope of possible configurations and the ease with which they're done is a hallmark of the best SaaS HRM software. Incorporating such a high degree of configuration capability within a multi-tenant product does require more architectural sophistication than doing so within a traditional single tenant product, but it is certainly doable as evidenced by how many of the best current SaaS vendors have accomplished this.

Whether you're an HRM BPO provider or SaaS vendor, your revenues don't really hit until that customer is implemented/migrated, so you have a profound interest inshortening the elapsed time, therefore the configuration time, to get that customer into production. And whether you're a BPO or SaaS customer, you're going to bear the cost of the implementation/migration as well as not get the benefit until you're fully in production, so you have a profound interest not only in shortening the elapsed time but also in reducing the level of effort, errors, and change management to get you there. So, even as robust, effective-dated configuration tools reduce greatly the upfront and downstream costs and risks of HRM software implementation/migration, there's still a lot of work to be done which could, potentially, be automated further. Enter next month's column: interrogatory configuration. HRO

Naomi Lee Bloom, Managing Partner, Bloom & Wallace, can be reached at 239-454-7305 or naomibloom@mindspring.com. You can also follow her on Twitter @InFullBloomUS or on her blog http://infullbloom.us