The Tower of Babel in HRM

Successfully outsourcing your HRM processes can seem like an undertaking of epic proportions. And if you aren't all speaking the same language, your tower (or your outsourcing) could crumble into chaos. By Naomi Bloom

Naomi Lee Bloom, managing partner of Fort Myers, Florida-based Bloom & Wallace, may be reached at 239-454-7305 or naomibloom@ mindspring.com.

One of the biggest challenges in any attempt to apply information technology to human resource management (HRM) is that we don't yet have a vocabulary for discussing major concepts, let alone details. Is my candidate your applicant? Is my compensation plan your rewards program? Where does compensation end and benefits begin? Does your new hire transaction include all of the needed onboarding processes? And what on earth do "end-to-end offerings" encompass? Without clear and consistent definitions for HRM concepts, events, data, processes, and metrics, it's little wonder that comparing HRM software and outsourcing offerings is a challenge.

If you've found yourself in this dilemma, let me offer a solution: Develop a "domain model"—a formal description of the HRM domain, complete with clear and consistent terminology and definitions. A domain model describes the domain (the subject matter under consideration) from the perspective of the business events, data, processes, metrics, and business outcomes relating to it.

Colleagues in financial management have a leg up on understanding domain models, which is why there are few jokes about CFOs misunderstanding the term "expenditure." For those with different backgrounds, consider this: software experts understand the clarity needed for an HRM object model; HR subject-matter experts understand the clarity needed for accurate reporting (e.g., what counts as a "head" in a headcount?); and outsourcing providers try to achieve the same clarity every time they write a service level agreement (SLA).

In a perfect world, an HRM outsourcing domain model would be developed, vetted, and provided "free" to all by a professional association. In that same perfect world, I would be twenty-five again and know what I know now. In the real world, every HRM software vendor and outsourcing provider offers their customers a vision of their own corner of the domain, which may not agree with the object models on which their software is based (or anyone else's.) Moreover, many of these domain visions are at such a high level as to make any meaningful discussion about them next to impossible. Does it address important issues such as whether insurance-carrier selection is included in the benefits administration offering or whether payroll services include financial accounting for payroll?

Just because the sales team may not present the provider's domain model in detail doesn't mean that the provider doesn't have a much more useful one. At a minimum, they are using software as part of their HRM delivery system (HRMDS) that has a physical object model at its core. Where multiple pieces of HRM software have been tied together to create their HRMDS, there should be a logical domain model that plans for the needed systems integration work. The best providers will have invested in developing a complete domain model, at least at the highest levels, and then go much deeper in the areas they are planning to cover in their service offering(s). And you should certainly expect that any SLA will be written with their domain model's precise terms and definitions for everything that they plan to offer, along with the related metrics.

In the early days of packaged software, I advised clients to request the vendor's data model as the best way to quickly see the vendor's assumptions about the HRM domain and to test how well those assumptions would fit the customer's needs. If there was no person object, there was certainly no way to distinguish properly between persons who were employees and those who were non-employee workers. If there was a one-to-one relationship between an employee and a position, then there was no way to support the common practice of having an individual work two part-time positions in order to secure full-time benefits.

Needless to say, I was bludgeoned by the software vendors for exposing the extent to which their object models revealed the limitations of their assumptions and, ultimately, their software. Aging and poorly conceived data/object models have contributed mightily to the high cost of ownership of in-sourced software, initially and over time.

I expect that some of the outsourcing providers will be equally chagrined if their prospects start asking them for their domain model, but those models tell prospective outsourcers everything about that provider's understanding of and assumptions about the domain—as well as how cost-effectively they'll be able to support that domain over time. Good domain models lead to good service offerings, good HRMDS platform designs or selections, clear SLAs, and many other positive outcomes. Poor domain models or worse, no domain models, produce just as poor results for outsourcers as they did for software vendors. HRO