IT Think Paper #17

**ERP versus Best of Breed**

**Edward G. Happ, Global CIO**

Consideration of an Enterprise Resource Planning system (ERP, for short) is a natural event for a National Society or Nonprofit at a certain stage of development. We at the IFRC Secretariat went through that phase some years ago when SAP was introduced via our HR system upgrade, rather expensively and with questionable success. IFRC was not alone in its experience. INGOs such as CARE, Save the Children, Children International and UNICEF have had similar experiences with varying degrees of success and shortfalls. Why are such run-the-business systems so difficult? Here are some reasons to consider[[1]](#footnote-1):

1. Too expensive. UNICEF spent an estimated $20M+ to roll out SAP to their 120 country offices. Even limited ERP projects, with nonprofit discounts, take $2M+ to complete. 80-90% of the costs are for business process and implementation consulting.
2. Takes too long. The typical ERP project at large NGOs takes 2.5 years or more to complete. For 5-year strategic plans it means that the impact of the new system is limited. Save the Children spent 3+ years to complete its donor ERP, Blackbaud's BBEC, missing its strategy window. Confounding this is that technology changes and user needs are evolving faster than 3-year ERP development cycles, ensuring that the system once released will likely be out of date.
3. Failure rate is too high. According to the well-known Standish Chaos database, fully two-thirds of large IT projects fail. One-third fail outright, while one-third are over time or budget.[[2]](#footnote-2) A recent McKinsey report verifies this[[3]](#footnote-3). Smaller, chunked-down IT projects have higher success rates.
4. Don't meet financial objectives. The Nucleus ROI study found 57% of large SAP projects (a leading ERP) don’t achieve the ROI cited in the beginning as the justification for doing the project in the first place.[[4]](#footnote-4) Furthermore, an analysis of 100 corporations that have implemented SAP shows that this group has significantly lower profits than peers.[[5]](#footnote-5)
5. Too hard to change. The pain and effort to upgrade business processes and customize the ERP system results in a system that is resistant to change, in terms of time, cost and will. As a result the ERPs that often replaces a legacy system (or three), becomes itself a new legacy system.
6. Not optimized for the web. Many of the large ERP systems, like SAP, have not been optimized for the web and browser world. Web features have often been bolted on to antiquated back-office architectures. Web users notice the difference from popular web apps immediately.
7. Almost impossible to please all departments. Given the enterprise nature of ERP systems, they cut across many departments. Getting many departments to agree on features and functions often means a tendency to lower common denominators of need rather than an optimal solution for an individual department needs. And yet each department wants its needs met, which often results in a high degree of customization.
8. Expensive to customize. The short-term cost to customize large ERP systems is high, with as much as 50% of the project costs going to the analysis, definition and development of components. However, the longer-term cost may be even higher, especially if the customizations are many. Each upgrade to the system that the vendor provides will most likely require an upgrade to the customized components, at the organization’s cost. This often results in a lagging behind the current market system, which gets worse over time.
9. User satisfaction is lower than for SaaS applications. Applications that are built for a web-centered world are more likely to meet user’s expectations for how modern applications should work. Users increasingly come to the organization with a strong experience base of using web-based applications in their education and personal lives. ERP systems often do not measure up and don’t function in a browser as users expect them to work.
10. We have a poor record of implementing large systems at IFRC. The HR SAP project, HLS Logistics project and Web Reconfiguration projects are all cases in point of large applications that were over-budget and over-time estimates by significant margins. One could take the approach of investing more in getting big systems right, or one could take the approach of chunking projects down to more fit-for-purpose solutions. We have chosen the latter.
11. Not share-able with National Societies. For ERPs to be shareable across organizations, their customizations often need to be repeated for different country/location needs. There is no multi-tenant model for ERPs that is comparable to SaaS applications, where each feature enhancement is made to the core code available to all.
12. Total cost of ownership (TCO) of ERP’s is affordable by only the largest organizations. For-profit organizations typically have 5 times more IT spending power per employee than large nonprofits[[6]](#footnote-6). With 25% of systems cost as a typical annual maintenance and support estimate, a $1M system can generate $250K in recurring costs, and this figure rises with the degree of customization. Over five years, the TCO can more than double the cost of a project. My advocacy position--with a bit of intentional hyperbole-- is that traditional ERPs will bankrupt Nonprofits and National Societies, customized ERPs even more so.
13. The cost per transaction is too high for a low volume enterprise. For an organization that has less than 60 emergency relief purchase orders per month, the cost per order when looking at a 5-year TCO can easily exceed $15K.
14. The data integration model of ERPs is overrated. One of the strongest selling features of an ERP is the shared database that underlies the system, providing the highest amount of systems and data integration. However, the real-time integration needs of many organizations are rare. If the data definitions are clear, the transfer of data among smaller applications is usually small enough for infrequent transfers of information. Sometimes a manual transfer of data is sufficient. Often a simple export and import of data will do. One-way integration may be needed. Two-way integration should be avoided. The key questions are: how often does the data change, how much of the data changes, and how many users are impacted. *A small scope demands a small solution.*

As a result of these considerations, the strategic direction for applications at IFRC are for small, fit-for-purpose, web-based, loosely coupled applications. A more agile organization is built on systems that are quick and easy to change, *and* faster to deliver. Those applications that meet a specific need are more likely to succeed. We believe this is the most prudent use of our donors funding.

1. A number of these points are developed in the unpublished paper I developed during my Executive Fellowship at Tuck/Dartmouth in June, 2008; the paper, titled “The Good Enough Principle – What we can learn about technology from the pragmatic solutions of nonprofits,” on [my web site](http://www.eghapp.com/hpmd/EGHprofile.nsf/links/50A6) (see the section under 2008). [↑](#footnote-ref-1)
2. “IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices,” R. Ryan Nelson, *MIS Quarterly Executive* Vol. 6 No. 2, June 2007, p. 67. Nelson notes that the Standish Group reports that “roughly two out of three IT projects are considered to be failures (suffering from total failure, cost overruns, time overruns, or a rollout with fewer features or functions than promised).” Also see the summary at:

   <http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS> [↑](#footnote-ref-2)
3. Their report notes: “Our research, conducted in collaboration with the University of Oxford, suggests that half of all large IT projects—defined as those with initial price tags exceeding $15 million—massively blow their budgets. On average, large IT projects run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted. Software projects run the highest risk of cost and schedule overruns.” [*Delivering large-scale IT projects on time, on budget, and on value,*](http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value)McKinsey & Company, October 2012, [↑](#footnote-ref-3)
4. SAP is the leading Enterprise Resource Planning (ERP) system adopted by many corporations to run the operations of most departments. "Fifty-seven percent of SAP customers interviewed did not believe that they had achieved a positive ROI, after having used their SAP applications for an average of 2.8 years. Those who did achieve a positive ROI limited customization and project scope, and they focused on user adoption and repeatability." Nucleus Research, March 2003 (see <http://nucleusresearch.com/research/notes-and-reports/the-real-roi-from-sap/> ) [↑](#footnote-ref-4)
5. "An analysis of nearly 100 public companies listed on SAP's Web site finds these SAP users are 20 percent less profitable than their peers. Despite SAP advertising claims to the contrary, factual analysis of ROE data shows the best run businesses don't run SAP." Nucleus Research, March 2006 (see <http://nucleusresearch.com/research/notes-and-reports/research-note-sap-customers-are-20-percent-less-profitable-than-their-peers/> ) [↑](#footnote-ref-5)
6. A student study conducted at Tuck/Dartmouth, for which I was an advisor, shows that large nonprofits spend a fifth of what for profit corporations spend. Cf. Hadley Fuller, Brad Lang, Aaron Mihaly, Kate Ryan Reiling, & Lisa Rockefeller, “Nonprofit Technology Needs Assessment & Guide, For Upper Valley Nonprofits,” May 21, 2008, for Prof. John Vogel’s Social Entrepreneurship class at the Tuck School of Business at Dartmouth. [↑](#footnote-ref-6)