

Chapter 69

A Decision Making Paradigm for Software Development in Libraries

Harish Maringanti

University of Utah, USA

ABSTRACT

Framing a technology question as a simple choice between developing an in-house application system and off-the-shelf proprietary system, or simply put, as a choice between build and buy, runs the risk of ignoring myriad options available in between the two extremes. In this era of cloud computing and run anything-as-a-service model, the very notion of developing an in-house application would raise a few eyebrows among C-level executives. How then can academic libraries, under mounting pressure to demonstrate their value (Oakleaf, 2010), justify investments in software development in particular? What follows in these sections is a brief discussion on the importance of investing in software development in libraries, three mini-case studies demonstrating the wide possibilities of integrating software development in library operations and a non-prescriptive model to assess which projects may be worth pursuing from the software development standpoint.

INTRODUCTION

Framing a technology question as a simple choice between developing an in-house application system and off-the-shelf proprietary system, or simply put, as a choice between build and buy, runs the risk of ignoring myriad options available in between the two extremes. As Langer puts it – “Choices of whether to make or buy do not necessarily need to be binary, that is, one or the other, but rather could end up as a hybrid decision. For example, an organization can develop its own application using open source within its application development strategy or it can license a third-party product that also contains open source.” (Langer, 2016). In this era of cloud computing and run anything-as-a-service model, the very notion of developing an in-house application would raise a few eyebrows among C-level execu-

DOI: 10.4018/978-1-6684-3702-5.ch069

tives. How then can academic libraries, under mounting pressure to demonstrate their value (Oakleaf, 2010), justify investments in software development in particular? What follows in these sections is a brief discussion on the importance of investing in software development in libraries, three mini-case studies demonstrating the wide possibilities of integrating software development in library operations and a non-prescriptive decision making paradigm to assess which projects may be worth pursuing from the software development standpoint.

Why Invest in Software Development?

Traditionally, businesses looked at IT systems as “black boxes” where it really did not matter what the interior looked like as long as those systems were able to deliver the value to the customer. Libraries were no different - one look at the literature points to a rich history behind library automation where proprietary systems played a major role (Breeding, 2006; Andrews, 2007). In the initial years, the focus was on automation where libraries cared about automating as many internal operations as possible. And as the years progressed, the expectations from such systems only increased to a point where automation alone wasn't sufficient anymore - integration between the automated components was expected. It can be seen in the natural progression of the vernacular from Automated Library System (ALS) to the Integrated Library System (ILS) (Kinner, & Rigda, 2009). Somewhere along, a big “mental shift” happened in libraries' thinking - the realization that technology could be leveraged not only to optimize the internal operations but also to drive innovation and deliver value to the users. As libraries demanded more of the vendors, the vendors responded by opening up the “black boxes” to give libraries some control over what and how things could be developed. Libraries, in turn, began responding to the local needs by hacking and developing tools on top of ILSes, where feasible.

But, libraries' future is not just in ILSes anymore - a remarkable array of new services are being developed. From generic institutional repositories to specialized subject-based data repositories, from offering mediated access to commodity collections to opening up access to troves of unique special collections - in every facet of the new and emerging programs, one can see the influence of technology driving the future. Though vendors have been great collaborators and partners in working with the libraries to support and sustain traditional library operations, it is imperative for the libraries to take control of their future and make investments in technology to support new and emerging services. Tyler Walters and Katherine, urge research libraries to “Be a doer, not a broker” and to “bring these services in-house (e-publishing services, hosting and curating digital archives, datasets) and seize the moments of opportunity as they arise” (Walters, & Skinner, 2011). Another important way in which developers (within libraries) can be a great asset is by looking out for solutions to solve what were once traditional library problems. Dan Chudnov expressed this sentiment eloquently - “The very notion of what has defined “library automation” for the past 30 years will shrink to mean primarily legacy operations; more and more of our critical systems, standards, and services will come to be supported by solutions developed outside of our community, and customized and optimized within it” (Gordon, 2007). Libraries need to invest in software development activities so that employees can implement this “outside-in” strategy to bring in solutions to solve current library problems from other fields.

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/a-decision-making-paradigm-for-software-development-in-libraries/294526

Related Content

Image Encryption using different types of Chaos-Maps and their Comparison

(2022). *International Journal of Systems and Software Security and Protection* (pp. 0-0).

www.irma-international.org/article//315584

Cooperation Patterns and Adaptation Patterns for Service-Based Inter-Organizational Workflows

Boukhedouma Saida, Oussalah Mourad, Alimazighi Zaiaand Tamzalit Dalila (2014). *Uncovering Essential Software Artifacts through Business Process Archeology* (pp. 250-283).

www.irma-international.org/chapter/cooperation-patterns-and-adaptation-patterns-for-service-based-inter-organizational-workflows/96624

An Empirical Study of the Effect of Design Patterns on Class Structural Quality

Liguo Yuand Srini Ramaswamy (2014). *Handbook of Research on Emerging Advancements and Technologies in Software Engineering* (pp. 106-125).

www.irma-international.org/chapter/an-empirical-study-of-the-effect-of-design-patterns-on-class-structural-quality/108613

An Investigation on Quality Perspective of Software Functional Artifacts

Vimaladevi M.and Zayaraz G. (2020). *Crowdsourcing and Probabilistic Decision-Making in Software Engineering: Emerging Research and Opportunities* (pp. 109-133).

www.irma-international.org/chapter/an-investigation-on-quality-perspective-of-software-functional-artifacts/235765

Fault Recognition for Mechanical Arm by Using Relative Margin SVM

Dongzhe Yang (2022). *International Journal of Information System Modeling and Design* (pp. 1-10).

www.irma-international.org/article/fault-recognition-for-mechanical-arm-by-using-relative-margin-svm/313576