Software Services: A Research Roadmap

-R.Vamshi Krishna and S.Sunil Kumar

This article focusses on four selected topics:

1. How to organize people and the flow of work through people
2. How to manage knowledge at an organizational level
3. How to estimate and manage risk in a services engagement.
4. Testing services.

And also describes about the authors experience in deployment of research innovations in a large service delivery organization

* Most businesses prefer to focus on their core competence, rather than develop and foster in-house competence for software development and significant customization and systems integration is required to run any large enterprise, even if substantial building blocks are available off the shelf are the main drivers for demand for software services.

* Quality of service informally means several things that one may associate with any kind of services business, even outside of information technology can the service provider deliver on their contract on time, with acceptable work quality, and at a competitive price.

* Some topics of current prominence, that are not considered in this paper, include legacy transformation, API identification and extraction, and migration of applications to cloud.

* Software services are of three kinds:

- Consulting refers to business-level consulting.

- Systems integration refers to assembling a solution from off-the-shelf software and hardware components.

- Application services refer to service engagement.

* To differentiate, Show the innovations that they bring to the table that can help the health-care management company in terms of cost or quality are the necessary.

* Implementation and testing phases resemble a development or a legacy transformation project at a product company. Finally, the transformed portfolio enters steady-state maintenance phase, where field bugs are handled and customer support is provided.

* Collaborative development platforms that promote structured interaction between team members can help make distributed development more efficient.

* Adoption of cloud and API-centric approaches to development and agile methods are also helping to reduce the inefficiencies inherent in distributed development.

* Advantages of the competency-center approach are resource multiplexing as well as specialization.

* Competency-center model has now become central to many IT vendors and large enterprises, it has seen less interest in literature as the focus is on large-scale development with a global workforce, which is hard to replicate in an academic setting.

* Five problems can significantly improve the performance of these competency centers

P1. Work envelopes: Central to the concept of a competency center is the ability to partition software projects into granular work envelopes.

P2. Measurement: Information reported by the practitioners of a competency center is often unreliable due to the complex structure of incentives involved.

P3. Estimation: Competency centers deal with large volumes of repeatable tasks belonging to a few categories that are performed by dedicated teams. Typically reduces the variance in the estimated time required to perform these tasks.

P4. Governance: Enterprises have extended traditional project management techniques to include competency centers.

P5. Planning and scheduling: Traditional calendar-day project planning is highly inefficient in the context of a competency center. A better approach would be to use a queue-based model, where the queue is managed centrally to re-prioritize the tasks in a practitioner's queue to ensure on-time delivery.

* Consumers can use the enabler to ascertain the risks that in sourcing from a particular provider. Providers in turn get the ability to price their services competitively based on delivery record.

* For traditional IT vendors - conceptual services marketplace helps offload low margin IT services to smaller vendors with less overhead.

* For customers - conceptual services marketplace provides an opportunity to outsource small work without entering into expensive long-term contracts.

* For the marketplace providers - a role will be played by traditional IT vendors. Conceptual services marketplace is a chance to profit from helping both providers and consumers benefit from the marketplace by offering some guarantees.

* Authors anticipate that the service requests in such a marketplace will not be full-fledged software projects, but of the size that can be performed in a few days or weeks by a small team of developers or even individuals.

* Authors proposed a conceptual illustration of a distributed services marketplace which is an extremely important.

* A knowledge-management system is needed to create project memories that can serve different needs, such as assisting new team members in identifying experts to reach out to for specific questions, or helping existing team members determine the relevant artifacts.

* Store the collective knowledge of past engagements, processes, and people to increase productivity and reduce activities that "reinvent the wheel." are essential for companies to build organization memories.

* There are challenges in:

(1) knowledge creation—how to codify explicit and tacit knowledge and motivate individuals to contribute.

(2) knowledge retrieval—data versus information versus knowledge;

(3) knowledge governance— legitimacy, relevance, and quality of contributed knowledge.

* Troubleshooting, Software Development Projects, Service improvements are three typical scenarios in service delivery that can benefit from an OKMS.

* Crawling, Parsing and annotating are basic aspects of knowledge creation.

* Indexing and Searching, Search result organization, Advanced analytics are the aspects of knowledge retrieval.

* Software risk management is a well-established discipline that has generated continued academic interest as the complexity and nature of the software projects have evolved overtime.

* Custom development for individual customers is more challenging and, despite best efforts, dramatic failures are quite common in large custom software-development projects.

* For the testing services, authors implemented a life cycle of test plan organization, test data generation, test automation, test execution, regression testing and test maintenance.

* From the authors past experience of taking research innovations from the lab to real-world deployment in IBM Global Business Services they concluded that any new process or tool causes at least a minor disruption in normal flow of the project, and this makes pushing either a new process or a new tool into project delivery significantly harder.

* Finally, the authors concluded that software services companies rely crucially innovations in individual and organizations productivity for their competitiveness and research community needs to take a careful look at the software services industries.