

Systems Analysis and Design in an Age of Options

G. Spurrier and H. Topi

Chapter 7—Identifying Development Options:

Selecting Implementation Approach and Determining Sources of Resources

© Prospect Press

Chapter 7: Identifying Development Options: Selecting Implementation Approach and Determining Sources of Resources

• • •

Systems Analysis and Design in an Age of Options

Learning Objectives

- Recognize multiple, often integrated, options to achieve project objectives: COTS, SaaS/cloud-based service, open source, new development
- Identify multiple sources of development resources: internal staff, contractors, consultants; domestic, nearshoring, offshoring
- Understand the benefits and disadvantages of various development approaches and sourcing options

Learning Objectives, Continued

- Select externally developed software capabilities; work with vendors
- Understand the pros and cons of different contracting approaches, ranging from time and materials to fixed-scope/fixed-cost contracts, and approaches in between those extremes
- Select consultant(s) for a project

Key Questions about the Sourcing of Systems Capabilities

- Which elements of the system are specifically developed for the user organization? Which have been or will be developed to be shared?
 - Buy vs. build
- Who employs the experts who design, develop, and/or configure the system?
 - Outsourcing vs. internal staff
- Where are the contributing experts located?

Systems Development Process Framework

Systems Development Process Framework: Hybrid Approach

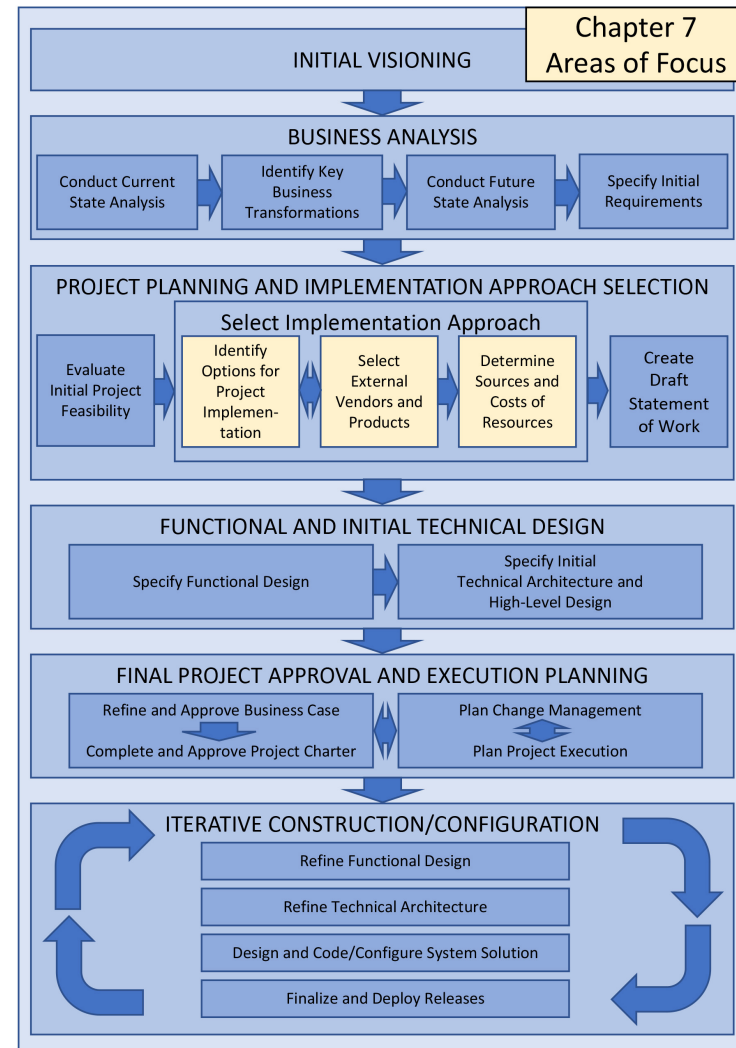


Figure 7-1 Selection of implementation approach in the context of the Systems Development Framework

Approaches to Acquiring Software Capabilities

- Development of custom, proprietary software solutions
- Licensing commercially available software solution
- Open source software (OSS)
- Integration of the above

Development of Custom, Proprietary Software Solutions

- Justification for use
 - Achieving distinctive (or even unique) organizational capabilities with the systems based on the software to support a unique mission and strategy
 - Collecting forms of customer data that competitors do not have access to
 - Desired capabilities simply are not available from the market
 - In certain circumstances, internal development is cheaper

Development of Custom, Proprietary Software Solutions

- Costs and Risks
 - Often expensive
 - Long-term commitment to keep maintenance staff and react quickly to environmental changes (such as legal and regulatory changes)
 - Inherent risks associated with creating new software
 - Ability to hire competent staff

Licensing Commercially Available Software Solutions

- Three categories:
 - Large-scale enterprise systems: ERP, CRM, SCM
 - Specialized systems for industry verticals
 - Traditional packaged software
- Delivery modes:
 - On-premises (on-prem)
 - Cloud-based infrastructure
 - Software-as-a-service (SaaS)

Types of COTS Solutions

Infra-structure \ COTS type	Enterprise systems	Specialized solutions	Traditional packaged software
On premises			
Cloud-based/IaaS			
Cloud-based/SaaS	↓	↓	↓

Figure 7-3 Types of COTS solutions. The arrows indicate movement toward cloud-based SaaS solutions.

Licensing Commercially Available Software Solutions

- Activities during implementation and maintenance
 - Construction (CON)
 - Change management (CM)
 - Deployment (DEP)
 - Glue code (GC)
 - Maintenance (MAN)

ERP Implementation Activities

Construction/configuration (CON)	
	Tailoring
	Configuration
Glue code (GC)	
	Integration of component applications and systems
Change management (CM)	
	Business process changes
	Business process documentation
	Training
	Support mechanisms
Deployment (DEP)	
	Transition period
	Transferring data to the new system
	Managing business process transition
Maintenance (MAN)	
	Managing relationships with vendors
	Release management

Figure 7-4 ERP implementation activities

Licensing Commercially Available Software Solutions

- Justification for Use
 - Shared development and maintenance work should lead to shared, and therefore lower, costs
 - Costs are more predictable
 - Lower general risk because of prior users
 - Faster access to functioning system
 - Desire to get access to “best practices”
 - But are they genuinely the best if everybody has access to them?
 - Perceived competitive necessity

Licensing Commercially Available Software Solutions

- Costs
 - Non-trivial acquisition and ongoing licensing costs + costs for hiring a competent implementation partner
 - Shared best practices might end up being a road to mediocrity
 - A large-scale enterprise system might create an incentive to increase complexity unnecessarily
 - Known usability challenges
 - Dependence on the vendor and very high switching costs

Open Source Software

- Available without direct monetary cost, including source code
- Modifications need to be made available to others for free
- Widely successful in the context of systems software:
 - Operating systems: Linux, other open source Unix variants
 - Data & Analytics: MySQL, MariaDB, Hadoop, MongoDB, Redis, Apache Spark
 - Apache web server
- Broadly used software development tools and environments, such as
 - PHP, Python, Eclipse, Ruby on Rails, Node.JS, etc.
- Some application software packages have become relatively widely used in their specialized categories
 - RapidMiner, Sakai

Practical Reality: Integrated Systems

- In the great majority of organizations, any new system capabilities have to be integrated with existing systems or they will not be effective
- The application portfolio of most modern organizations consists of software applications from a variety of different sources
- Integration of different components with a variety of mechanisms, such as glue code

Integration of Unique and Shared Functionality

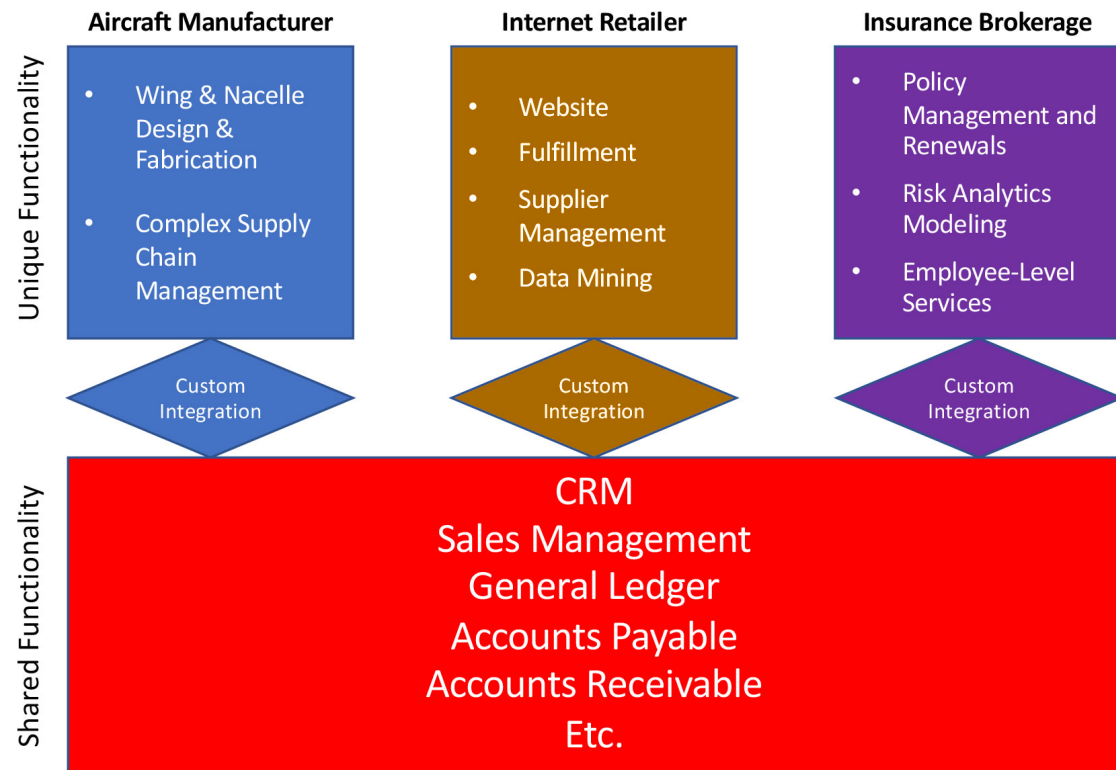


Figure 7-2 Examples of systems with unique and shared functionality

It Is Essential to Maintain

- Control and understanding of the overall architecture of the solution
- In-depth understanding of the technical mechanisms used for integration
- Strong relationships with the vendors of all components so that changes to the component systems do not cause avoidable surprises
- The ability to react in a flexible way to unavoidable surprises

Sourcing of Development Resources

- Two major questions
 - Type of contractual arrangement
 - Geographic placement of resources

Types of Contractual Arrangement

- Hiring salaried employees
- Hiring independent contractors
- Acquiring contractors through a professional services firm
- Outsourcing development activities
- Using consulting services

Hiring Salaried Employees

- Directly employed by the company that is responsible for developing the software capabilities
- Advantages
 - Stability of workforce in general
 - Stability of teams
 - Learning to know staff well
 - Developing a good understanding of what teams can achieve
 - Predictable staff availability
 - Employee loyalty
- Disadvantages
 - Lack of flexibility
 - Inability to react quickly to technology changes
 - Potentially more expensive
 - Narrow focus on specific projects → skills gap

Hiring Independent Contractors

- Individual professionals who provide their services in a series of short-term assignments
- Not employees of the organization developing the software
- Essential to determine the right way to differentiate between contractors and employees (major legal ramifications possible)
 - In the U.S., if the employer has “the legal right to control the details of how the services are performed,” the individual is an employee, not a contractor (<https://www.irs.gov/businesses/small-businesses-self-employed/independent-contractor-defined>)
- Advantages: Flexibility, well-defined total costs, ability to acquire a resource for a short-term need
- Disadvantages: Missing benefits of long-term employment, leading to lack of loyalty, lower team stability and ability to learn as a team, higher turnover costs, etc.

Contracting Developers through a Professional Services Firm

- Essentially the same as hiring independent contractors directly, with the important exceptions of:
 - Professional services firm is clearly the employer
 - Unit costs are higher

Outsourcing

- Larger segments of IT work are contracted to a service provider as an alternative to its own employees performing the work
- Governed by service-level agreements
- A broad range of capabilities can be outsourced: infrastructure, maintenance, development, analytics, etc.
- In development, it is typical to outsource maintenance and continuing development of older parts of the application portfolio so that the company can focus on new development
- Well-known firms: Accenture, TCS, Cognizant, Wipro, IBM, HCL, Infosys, Capgemini, DXC, NTT Data

Consulting

- Outsourcing focuses on providing specific services based on an SLA; in consulting, the emphasis is more on guidance regarding performance improvement and future-focused capabilities
- More open-ended and less tightly specified than outsourcing
- A larger part of the overall IT team is external than in outsourcing

Geographic Placement of Resources

- On-site
 - Physically co-located teams working together
- Onshore outsourcing
 - Outsourcing to providers within the client organization's home country
- Nearshoring
 - Offshoring to locations that are geographically close to the client's home country
- Offshoring
 - Outsourcing to vendor outside the client organization's home country (in practice, often either geographically or culturally distant)
- Captive units
 - An internal organizational unit located geographically outside an organization's typical operating area

Current Reality: Geographically Distributed Teams with Multiple Organizational Arrangements

- Advantages:
 - Ability to benefit from lower labor costs
 - Access to larger pools of resources
 - Broadening the organization's understanding of cultural and political environments
 - Development work can continue without interruptions
- Disadvantages:
 - Lack of face-to-face (or any type of synchronous) communication
 - Creation of trusting personal relationships more difficult
 - Functional specifications become essential but are more difficult to write
 - Higher coordination costs
 - Potential security risks

Managing Vendor Relationships

- Selecting a vendor
- Types of contracts
- Building a long-term relationship with a vendor

Selecting a Vendor

- Process
 - determination that a third-party solution should be used via selection and configuration (rather than developing new software via construction)
 - identification of potential vendor candidates
 - articulation of the required services in a request for proposal (RFP) or request for information (RFI) document
 - initial evaluation of the received proposals, leading to a selection of short-listed candidates
 - presentations and product demonstrations by the vendors on the short list; and
 - selection of the winner

Selecting a Vendor

- Criteria
 - Quality, fit, and growth paths of the solution architecture
 - Vendor's reputation and financial viability
 - Vendor's support capabilities
 - Cultural fit
 - Vendor's approach to tailored solutions and version management
 - Vendor's track record in implementation
 - Vendor's size and power compared to the potential client

Type of Contracts

- Application Software Development and Maintenance
 - Fixed Price
 - Time and Materials
- Enterprise systems and other third-party software
- Cloud infrastructure for systems development and deployment

Chapter Summary

...

Let's Review

- Three key questions:
 - Which elements of the system are specifically developed for the user organization? Which have been or will be developed to be shared?
 - Who employs the experts who design, develop, and/or configure the system?
 - Where are the contributing experts located?
- Four key approaches to acquiring software capabilities:
 - Development of custom, proprietary software solutions
 - Licensing commercially available software solutions
 - Adopting open source software (OSS)
 - Integration of the above
- Licensing externally developed software still requires significant amount of time and resources
 - Construction, Change management, Deployment, Glue code, Maintenance
- Sourcing of software development resources
 - Hiring salaried employees
 - Hiring independent contractors
 - Acquiring contractors through a professional service firm
 - Outsource development activities
 - Use consulting services
- Geographic placement of resources: on-site, onshore outsourcing, nearshoring, offshoring, captive units

Let's Review

- Current reality: geographically distributed teams with multiple organizational arrangements
- Also in this area, we truly live in an era of options