Design for the New HR System - Riordan Manufacturing

Angie Ellis

BSA 375

November 4, 2013

Paul Cooper

**Introduction**

The Chief Operating Officer of Riordan Manufacturing has asked for a detailed system design and project implementation plan for a new system for the human resources (HR) department. He wants to take the existing HR tools that the company has and integrate them into one application using more sophisticated advanced technology that is available. The installation of their existing system, HRIS, happened back in 1992. It uses basic technologies, such as Excel spreadsheets to keep track of information. It is time to create a new, state or the art system for the HR department to optimize their productivity and support the success of the entire company.

The new system installation will take place at each Riordan plant location. The expected completion date is in approximately six months and the expected accessibility for use is in the second quarter of next year. The first step in the design for this new HR system for Riordan Manufacturing is systems analysis. This step requires understanding the keys to success in the analysis phase, defining the scope of the project, conducting feasibility studies, requirements gathering, and data modeling. (University of Phoenix, 2005)

**System Analysis**

**Keys to Success in the Analysis Phase**

Before we can begin systems analysis, we must understand the key factors that lead to a successful analysis phase in software development. First, we must understand thoroughly the problem leading to the creation of the new system before we gather the requirements. Second, we must involve the stakeholders from the beginning and continue to involve them throughout the development process. Third, we must consider the critical requirements. Fourth, we must give attention to nonfunctional and quality requirements, such as security, reliability, portability, etc. These are important factors to implement in our development of the new system for Riordan Manufacturing to optimize our chances for success. (Antón & Wells, 2003)

**Defining Project Scope**

A crucial part of the systems analysis phase is the defining of the new system’s scope (Valacich, George, & Hoffer, 2012). Defining the project’s scope will help the project manager and the customer align his or her ideas for the system. It will show an understanding between the two parties. The system’s scope also gives a reference point for control. By defining the project objectives, there will be a definitive statement to measure changes and success with (Kraus & Cressman, 1992).

To help define the scope, there are specific questions that need asking. First, we can ask which departments at Riordan could use the new system or could the new system affect. Second, we can ask which current systems at Riordan could change as a result of the new system or need to interact with the new system. Third, we can ask who the stakeholders are for the new system. Finally, we can ask what potential functionalities will exist in the new system. These questions will help define the project’s scope and help us understand the objectives the system needs to fulfill. (Valacich, George, & Hoffer, 2012)

**Conduct Feasibility Studies**

The next part of the design process is to conduct feasibility studies for the new system. This will tell us how the new system will affect the company and if it will be worth the time and resources to build it. There are three areas of feasibility that we must explore. These areas are economic, technical, and organizational.

**Economic feasibility.**

To determine economic feasibility, we need to complete a cost-benefit analysis for the new system at Riordan Manufacturing. A cost-benefit analysis will evaluate the financial risks and opportunities of the system. The financial risks include development costs, operating costs, and intangible costs. The financial opportunities include the monetary benefits to Riordan, potential cost savings, and intangible benefits. Performing an economic feasibility study will tell us if it will be worth it for Riordan to build the new HR system. (Dennis, Wixom, & Roth, 2012)

**Technical feasibility.**

To determine technical feasibility, we need to evaluate the possibility of the new system’s effective development and implementation from a technological standpoint. The issues to explore with technical feasibility are Riordan’s familiarity with the application and the technology with the new system. The lower the familiarity with these two aspects, the higher the risk. Project size and compatibility with existing systems are also a concern. The larger the project size, the higher the risk involved. The lower the compatibility with existing systems at Riordan, the higher the risk. Performing a technical feasibility study will tell us if it will be possible for Riordan to build the new system. (Dennis, Wixom, & Roth, 2012)

**Organizational feasibility.**

To determine organizational feasibility, we need to evaluate how well Riordan employees will accept the new system the how well the new system will incorporate at Riordan. We need to determine if stakeholders of the new system support the project and will accept it when it is complete. We also need to compare the objectives of the project with the business objectives of the company. The closer the objectives are in line, the lower the risk of the new system. Performing an organizational feasibility study will tell us if employees will accept the new system into the organization. (Dennis, Wixom, & Roth, 2012)

**Requirements Gathering**

The next part of the design process is requirements gathering. The main method of requirements gathering will be through Joint Application Design (JAD). There are significant advantages to using JAD for requirements gathering. First, JAD brings together several key people for one or more sessions of requirements gathering. With everyone together in a room at once, the discussion of different perspectives, ideas, and issues can happen to ensure thorough collection of system requirements. Second, JAD can greatly accelerate the development process by accomplishing the considerable task of requirements gathering in a short amount of time. Third, one of the main causes of project failure is incomplete or changing requirements. Studies of JAD have shown a lower incidence of errors in requirements gathering and, therefore, a higher chance of success for a project (Wood & Silver, 1995).

The key stakeholders who need to be at the requirements gathering sessions are the President and Chief Executive Officer, the Chief Operating Officer, the Chief Financial Officer, the Director of Human Resources, and the employees in the Human Resources department who will be using the new system. A role of the President and CEO, Dr. Michael Riordan, is to help formulate business strategies at Riordan. He could be an important asset to the system development. The COO, Hugh McCauley, was the original requestor of the new system for the HR department. A role of the CFO, Dale Edgel, is to direct functions of HR toward the goals of the company while following the values of the company at Riordan. A role of the Director of HR, Yvonne McMillan, is to direct and coordinate HR activities. Finally, the perspective of the HR employees who use the system will be valuable. Gaining the perspective and wisdom from each of these individuals will help determine the proper requirements for the new system.

**Data Modeling**

After successfully gathering the requirements for the new system, we will conduct the next part of the design process, which is data modeling. To model the data we have collected, we will use data flow diagrams, decision tables, and entity-relationship data models. Data flow diagrams are visual representations of the movement of data through systems. Data flow diagrams also show elements inside and outside the system, people and technologies used, processes employed, and inputs and outputs. First, we must draw a context data flow diagram. This diagram shows one process that represents the whole system and provides a general overview of the system. Next we must draw a data flow diagram of the current system. Then, we must draw a data flow diagram of the new system. Finally, the data flow diagram of the new system needs breaking down into diagrams of each component in the system. Once each of these is drawn up, the diagrams need comparing and checking for completeness, consistency, and efficiency. In areas where the logic becomes increasingly complex, we can use decision tables. Decision tables display process logic in tabular form, showing every possible condition and action.

Drawing the data flow diagrams will help us understand the behavior of data in the system, but it will not help us understand the definition, structure, and relationships within the data (Dennis, Wixom, & Roth, 2012). With this the conceptual data modeling comes in through entity-relationship (E-R) data models. We will create two E-R models. First, we will create an E-R model of the existing system at Riordan. Second, we will create an E-R model of the new system that includes the scope and requirements. These models will help us better understand the entities involved in the system, their properties, and the relationships between entities.

**System Design**

The next step in the design process for the new HR system is system design. While during the system analysis phase we decide what the system will do, during the system design phase we will decide how the system will do it. During this step, we will design each module of the program and explain them in detail and so the programmer can start coding directly from the deliverables (Amlani, 2012). Essentially, we will create a blueprint for the new system in this step. To create a successful system design, this step requires a system acquisition strategy, refinement of nonfunctional requirements, and design of the system architecture, including network, program, interface, and database architecture.

**System Acquisition Strategy**

The first part of the system design process includes deciding on a system acquisition strategy that properly will meet the requirements of the new system while considering feasibility. The options for system acquisition are developing a custom system in-house, purchasing a packaged system, or outsourcing.

**Custom in-house.**

The ideal option for Riordan Manufacturing would be to build a custom application in-house. This option would give developers the flexibility to create a highly specialized system tailored exactly to the needs of the Riordan HR department. Developing the system in-house will build the technical skills and knowledge of the IT employees at Riordan, making future projects easier. However, this option may not be feasible. Building a custom system requires a significant amount of time and dedication from a team of employees who must have the proper skills. The risks for building a custom system can be high. Also, the cost involved with building a system in-house can be very high. Because building a system in-house may not be the most feasible option, we need to explore other options.

**Packaged system.**

Another option for system acquisition is purchasing a packaged system. There are thousands of commercially available software programs that can meet several common business needs (Dennis, Wixom, & Roth, 2012). Packaged software is a good option for common business needs and companies that lack the skills to build a system in-house. This option can be less expensive and quick to set up at a company. However, packaged software does not always meet business needs as well as other options. Although some customization may be possible, it may not be enough to meet a company’s requirements. A company may have to alter the way employees conduct business to make packaged software more effective in meeting their business needs, but this may be more trouble than it is worth.

**Outsourcing.**

The third option for system acquisition is outsourcing. Outsourcing involves the hiring of an another company to create or supply the system (Dennis, Wixom, & Roth, 2012). Outsourcing firms supply, host, and manage the software applications. Some outsourcing firms offer custom systems development. Advantages of outsourcing are lower costs, quick setup time, and reduced investments for in-house IT. However, outsourcing could mean lower control over future development, possible compromise of confidential information, and possible development of skills outside the organization (Dennis, Wixom, & Roth, 2012).

**Refinement of Nonfunctional Requirements**

The next part of the system design process is to refine the nonfunctional requirements of the new system into more detailed requirements. There are four types of nonfunctional requirements that need refinement, including operational, performance, security, and cultural and political requirements (Dennis, Wixom, & Roth, 2012). Operational requirements describe the operating environment of the system and any changes that may happen to the environment in the future. Performance requirements need to specify the required performance level of the new system in a measurable and verifiable way. Security requirements should specify how to protect the system from disruption or harm. The security requirements should be in line with Riordan Manufacturing’s voluntary standards of the ISO 9000. Finally, cultural and political requirements need refinement. Cultural and political requirements need specifications for the each country the system will reside. For Riordan Manufacturing, this will be the United States and China.

**System Architecture**

The next part of the system design process is to design the system architecture. There are five specific architectures that need designing for the new system, including network, program, interface, and database architecture.

**Network architecture.**

For the new system, we need to examine the different network architectures to decide which will best suit Riordan. The most common network architecture used today is the client server architecture. This type of architecture distributes processing between the client and servers. It provides scalability, tasks have clear separation between the client and server, it can support different types of clients and servers, and in case of server failure, minimal processes are affected. Although the client server architecture provides many benefits, it is a complex architecture for programmers to write programs for. This architecture is also more complicated to update. Despite the disadvantages, the client-server architecture is a good option to explore for the new HR system for Riordan.

Other types of network architectures include server-based architecture, client-based architecture, virtualization, and cloud computing. Server-based architecture is an older model in which the server performs all processes. Most businesses have moved away from this architecture for more updated models, such as the client-server model. A client-based architecture uses the server for storage only and whereas the client performs everything else. Although this architecture may work well where there are only a few users, this will not work well for Riordan. The virtualization architecture permits the creation of multiple virtual devices on one system. This can be a good option for companies to optimize the utilization of resources. Riordan may want to use virtualization for some of their servers or clients in conjunction with the client-server model. Finally, cloud computing is a new type of architecture where an outside company processes everything remotely but can be accessed anywhere at any time. This is an option worth exploring to use for the new HR system for Riordan as it can lower costs significantly, provides resources as needed and only charges for resources used, and leaves much of the headache of running a system to an outside company.   
**Program architecture.**

After deciding on a network architecture, the program architecture needs designing using a top-down modular approach. First, the logical DFDs from the systems analysis phase need converting to physical DFDs to show the specifications for system building and how the system will work after construction. The physical DFDs should include how data stores, data flows, and processes need implementing with each component (Dennis, Wixom, & Roth, 2012). The physical DFDs provide a good reference for the creation of structure charts, which need completion next. The structure chart organizes the processes for the new system, showing the organization of each part of the program and the interaction with other parts of the program. Finally, the program specifications need writing. The program specifications should include comprehensive information on what each program module should include. They should also include basic information for each module, processes, inputs, outputs, and pseudocode (Dennis, Wixom, & Roth, 2012). The specifications must be as clear and thorough as possible for successful program translation. These deliverables are vitally important because they will communicate the program architecture of the new system to the programmers.

**Interface architecture.**

Constructing the interface architecture includes designing forms and reports and dialogues. First, designing forms and reports involves interaction with systems users to design them to be effective as possible. The information gathered from the systems analysis phase will help create a prototype of the different forms and reports needed in the system. After the development of the prototype, users need to evaluate each prototype and provide feedback on what changes need to happen. This process will involve several iterations to arrive at the finished prototype of each form and report.

The forms and reports should include as much consistency as possible. They should have standard formats that match a typical paper format at closely as possible. Data-entry and navigation should also be as consistent as possible. The forms and reports should also include feedback and help users.

Second, dialogues need designing by treating the dialogue like a conversation between two people. The dialogue design consists of a three-step process. First, the dialogue sequence needs designing. By taking the information gathered in the systems analysis phase, a formal dialogue sequence can be written. Second, a prototype of the dialogue needs building. Third, the usability of the dialogue sequence needs to be assessed by system users. This will also be an iterative process like the design of forms and reports. Also like the forms and reports design, the dialogues have maximum consistency in their design. (Valacich, George, & Hoffer, 2012)

**Database architecture.**  
Last, the database architecture of the system needs designing. To optimize the process of design databases for the system, we need to use a good computer-aided software engineering (CASE) modeling tool. Using a CASE tool, we need to create logical database designs for each interface for the application, which describes the structure of the information. Next the logical database designs need combining into one logical database model. From the logical database designs and E-R data models created in the systems analysis phase, we will create the physical database design, which defines how physically to store the data (Poolet, 2011). It will also include the technical specifications for the databases.

This database design process should use normalization principles. According to Valacich, George, and Hoffer (2012), normalization is “a way to build a data model that has the properties of simplicity, non-redundancy, and minimal maintenance.” Using normalization principles will help develop a more effective database architecture for the new system.

**Conclusion**

Using the information collected from the systems analysis phase, the system design can occur. The design process is a top-down refinement process that moves from a high-level view down to the physical details of the system. This refinement process will help us show the programmers how to build the system clearly and thoroughly. With the deliverables from the system design phase, we can advance to the system implementation phase.

**System Implementation**

With the system design complete, we can progress to implementing the new system at Riordan. The development team takes the system designs and converts them into a real, working system. This phase requires a significant amount of time, money, and manpower. Seven separate stages require completion during this step. To implement successfully the new HR system at Riordan Manufacturing, we must complete coding, testing, installation, documentation, training, support, and maintenance.

**Coding**

The first part of the system implementation process is coding. Before beginning the coding process, a programming language needs deciding on. Three things need consideration when choosing a programming language to use. First, the language must fit the purpose. For normal business application, Java or a similar language is appropriate, whereas specialized applications should use C++ or a similar language. Second, the choice of platform should help dictate the appropriate programming language to use. If other applications at Riordan use the same language the new system should use that language as well. Third, the skills of the programmers should also help dictate the appropriate programming language to use. Because no in-house programmers work at Riordan, this factor does not need consideration. (Britton, 2008)

After deciding on a programming language, the coding process can begin. The program specifications developed in previous phases contain detail for each program module that needs writing, including pseudocode. The programmers can take the program specifications and translate them directly into code for the new system. During the coding process, testing is also performed and proceeds in parallel (Valacich, George, & Hoffer, 2012).

**Testing**

The next part of the system implementation process is testing. The aim of testing is to find problems in the system and fix them to improve the quality of the product. Testing also helps ensure a level of confidence in the predictable behavior of the system in provided conditions (Tuteja, 2012). The testing process needs thorough planning before execution. For each test, a corresponding test case needs writing. Test cases should include a thorough description of the test case and the results. These need documenting not only for planning purposes but also for future testing of application revisions.

**Manual testing.**

The execution of the testing process for the new HR system at Riordan will include both manual and automated testing methods. The manual testing methods will include inspections, walkthroughs, and desk checking. Inspections involve a group of testers who compare the code to a list of common errors in that programming language. The walkthroughs involve the programmer walking members of the development team through their documentation to help them understand their methods and gather feedback. For desk checking, reviewers go through each line of code, ensuring solid logic for the program and checking for errors.

**Automated testing.**

We will use computers to complete the automated testing methods. The automated testing methods will include syntax checking, unit testing, integration testing, and system testing. Syntax checking uses a compiler to look for errors in the syntax. Unit testing executes each module of code alone to uncover any errors. Integration testing happens after unit testing and involves combining modules from unit testing and testing them together. This continues until all modules at all levels test with no errors. Finally, system testing happens after integration testing and involves integrating all the programs into systems.

**Acceptance testing.**

After the completion of manual and automated testing by the development team, the system must pass acceptance testing by users. Acceptance testing will include alpha and beta testing. Alpha testing involves the implementation of the system in a test environment in which it undergoes several tests by the development team. After alpha testing reaches the benchmarks set for it, beta testing occurs. Beta testing implements the system in the user’s environment with the user running the system using his or her own data. After passing beta testing, the new system is ready for installation.

**Installation**

The next part of the system implementation process is installation. There are four approaches most commonly used for the installation process, including direct, parallel, single location, and phased. The one most suited for the new system at Riordan is the single location approach. This approach involves implementing the new system at one company location whereas the old system continues to run at other locations. Once the new system runs properly at the one location, the company turns off the old system and installs the new system at all locations. This approach provides a lower risk and lower cost than other approaches.

Before installing the new system, the installation strategy need planning. First, the data needs conversion for compatibility with the new system. Second, system documentation needs updating with new system documentation. Third, any new hardware required for the system needs setting up. Finally, an installation schedule needs planning. Users need notification of the schedule well ahead of time.

**Documentation**

The next part of the system implementation process is documentation. The development of the system documentation, which includes details about a system’s specifications, began in the systems analysis phase and continued into the implementation phase. Now we must focus on the development of user documentation, which provides information to users on the system, how it works, and how to use it. Different types of user documentation need development to support properly system users. The main source of user documentation for the new system will be an online help system developed as part of an Electronic Performance Support System (EPSS). The online help part of the EPSS will include detailed information on how to perform tasks on the new system. Users should also receive printed quick reference guides in a small book format that include a short guide to basic system use. Other types of user documentation may need development, depending on the complexity and the number of users of the new system. Well-done user documentation can decrease costs and time associated with training and support.

**Training**

The next part of the system implementation process is training. Properly training employees to use the new system is a cost-effective way to increase productivity (Valacich, George, & Hoffer, 2012). The EPSS will also provide online training for users. Users can use the EPSS for training on their own time, at their own pace, and on their own computer. The development team will also teach formal classroom courses that users must attend. Finally, the development team will train a resident expert in the HR department that other users can turn to for help.

**Support**

The next part of the system implementation process is support. All forms of support need setting up before the system installation (Valacich, George, & Hoffer, 2012). Because the scope of the new HR system at Riordan is fairly small, only a few employees need to work to provide support for the new system. These employees need thorough training on the new system and have good people skills. Tools are available to help the support team manage system support. The use of an issue tracking system can help manage and maintain a list of issues that need resolving from the support team. Users should have the option to reach the support team via phone, e-mail, or instant messaging.

**Maintenance**

The final part of the system implementation process is maintenance. The effectiveness of maintenance needs measuring because maintenance can be so costly (Valacich, George, & Hoffer, 2012). System failures need logging and tracking over time to measure and evaluate maintenance activities. Another important part of maintenance is handling maintenance requests. Maintenance requests need categorizing and prioritizing. The maintenance team also needs to determine which maintenance requests need fulfilling and which do not. The management team should try to address maintenance requests in batches to reduce their workload. When the team addresses maintenance requests, they also need to decide when to release new, updated versions of the software.

Another important part of maintenance is configuration management. Configuration management ensures that the only changes made to the system are authorized changes (Valacich, George, & Hoffer, 2012). To help with configuration management, Riordan should set up a system library with a system librarian dedicated to keeping track of the baseline modules. Modules need to go through a check-in process of quality-control testing, documentation, and approval to enter the library. For those who need to make changes to the system, they need to check out a copy of the baseline modules through the system librarian. Before checking the modules back into the library, they need to go through another check-in process of quality-control testing and documentation. Maintenance and support activities should continue for the life of the system.

**Conclusion**

By following the tried and tested techniques of the Software Development Life Cycle (SDLC) for the design of the new HR system at Riordan Manufacturing, the chances of success for the system are significantly higher. Each phase of the SDLC followed in this paper use the most thorough methods of the SDLC to create the system to best fit the needs of Riordan’s HR department. Following the designs outlined in this paper will lead to the development of a successful system that will boost the productivity of the HR department and employees, leading to Riordan Manufacturing becoming closer to achieving their goals and the optimum success of their company.

References

Amlani, R. D. (2012). Elaborated Phases of Software Development Life Cycle. *International Journal of Computer Applications and Information Technology*, *1*(2), 1-4.

Antón, A., & Wells, D. (2003) Point/Counterpoint. IEEE Software, 20(3), 44-47. doi:10.1109/MS.2003.1196319

Ardagna, D., Francalanci, C., & Piuri, V. (2004). Designing and rightsizing the information system architecture. *Information Systems Frontiers, 6*(3), 229-245. Retrieved from http://search.proquest.com/docview/232056982?accountid=458

Britton, C. (2008, January). Choosing a Programming Language. *Microsoft Developer Network*. Retrieved from http://msdn.microsoft.com/en-us/library/cc168615.aspx

Dennis, A., Wixom, B. H., & Roth, R. M. (2012). *System Analysis and Design* (5th ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Kraus, W. E., & Cressman, K. R. (1992). Project scope definition: A practical approach. *Cost Engineering, 34*(12), 15. Retrieved from http://search.proquest.com/docview/220410301?accountid=458

Poolet, M. A. (2011). Database design essentials. *SQL Server Magazine, 13*, 14-18. Retrieved from http://search.proquest.com/docview/862325054?accountid=458

Tuteja, M. (2012, July). A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models. International Journal of Soft Computing and Engineering, 2(3), 251-257. Retrieved from http://www.ijsce.org/attachments/File/v2i3/C0761062312.pdf

University of Phoenix. (2005). Riordan Manufacturing [Multimedia]. Retrieved from University of Phoenix, BSA375 – Fundamentals of Business Systems Development website.

Valacich, J. S., George, J. F., & Hoffer, J. A. (2012). *Essentials of Systems Analysis and Design* (5th ed.). Upper Saddle River, NJ: Pearson Education, Inc.

Wood, J., & Silver, D. (1995). Joint Application Development (2nd ed.). Hoboken, NJ: John Wiley & Sons, Inc.