

System Evaluation Tables

Introduction

The more carefully your team examines options early in the system development life cycle, the easier it will be to make the difficult, critical choices that distinguish a successful project from a failure. Identifying, understanding and summarizing project priorities for functional and non-functional requirements for your project will enable your team and client to choose the best solutions.

Description

System Evaluation tables enable you to quickly and efficiently identify and summarize the advantages and disadvantages of various alternatives during any stage of the system development life cycle. For example,

- What is the best scope for the project?
- What is the best platform for implementation?
- Should we buy application software or create custom software or choose an open source solution?
- If we purchase application software, what is the best choice to meet project goals?

Purpose

System Evaluation tables facilitate communication between the project team and the client community. When difficult choices must be made to manage the scope of a project, create a system evaluation table to identify, summarize and clarify the alternatives available. This will enable the project team and the client to enter into purposeful discussions leading to successful decision making.

Creating a System Evaluation Table

To create your System Evaluation table start with the System Evaluation table template. Notice that the final column in Figure 1 is “Client Decision.” This allows users to create a fourth hybrid alternative that may emerge from the

discussion and best fits the user’s objectives and constraints.

If time is critical to the discussion, consider adding a simplified milestone summary as part of your evaluation. The team that created the System Evaluation Table in Figure 1 knew that they could not create the scope laid out by their client in the first meeting (Alternative 3 in their table). Instead of telling the client, “we can’t do all you are asking,” they created an Evaluation Table that offered three solutions. Two of the alternatives had advantages of providing an initial system with some functionality that could be completed by another team in the following semester. The third alternative had all the requested functionality, but could not fit into the semester time-frame. The milestone summaries for each alternative at the bottom of the table indicate finish times for Alternative 1 or Alternative 2 by the end of the semester, but Alternative 3 would require working into the summer—something the students would be glad to do on a paid internship basis.

Once the client and team have agreed upon an alternative, label the client’s choice as part of your documentation for your Project Charter. This is shown in Figure 2—the ICC client chose Alternative 1, an Open Source CRM created with CiviCRM. The client also asked the team to include social networking functionality. Their System Evaluation table indicates that the team agreed to add the social networking capability “if easy.” In fact the team was able to complete that additional functionality, but in the Project Charter, the System Evaluation Table made it clear that they were not guaranteeing that functionality as part of their original scope.

Benefits

By identifying and answering critical questions regarding alternate system solutions before beginning your functional build phases, you avoid having to redo design and construction work. A discussion of your list of advantages and disadvantages may unearth misconceptions that will save you hours of development time and reduce the total cost of ownership (TCO) for



Figure 1. System Evaluation Table with Milestone Summaries

	Alternative 1: Oracle back end w/simple reports	Alternative 2: Limited scope of data with prototype GUI	Alternative 3: Proceed with Initial Scope and requirements	Client Decision
Key Features	Good relational database model Test completeness with manufacturing sites and products	Prototype of portion of system	Oracle back-end with functioning GUI	
Advantages	Foundation for GUI implementation and future modifications Able to fit within time constraints of team	Prototype to show user groups Model data flow Show how system parts will be integrated	Functioning system	
Dis - advantages	Not fully complete No functioning front end	Incomplete back end Incomplete front end All elements of system not analyzed thoroughly	Will not fall within time constraints * Lack of continuity in development Phases may be cut short due to time constraints	
Questions	Will this be sufficient? How important is GUI design?	Which functions need to be demonstrated?	Could firm hire a part-time intern to start in March or April & continue team's work?	

Milestones

Alternative 1 Oracle back end w/simple reports

1. Gather and analyze all requirements
 2. Create E-R diagram of database relationships
 3. Implement E-R diagram
 4. Create queries & present for review of process
- leaves one-week slack until final deadline for course

Target Completion Date

- 3/12
- 3/26
- 4/23
- 5/7
- 5/12

Alternative 2 Limited scope of data with prototype GUI

1. Determine data to model
2. Gather requirements about data to be modeled
3. Implement database model
4. Design front end prototype
5. Code front end & present for review of process

Target Completion Date

- 3/12
- 3/26
- 4/17
- 4/23
- 5/7

Alternative 3 Proceed with initial scope and requirements

1. Gather and analyze all requirements
2. Create E-R diagram of database relationships
3. Implement E-R diagram
4. Create queries
5. Design front end displays and reports
6. Code front end, test, train program managers, install

Target Completion Date

- 3/12
- 3/26
- 4/23
- 5/7
- 5/21 (past end date)
- 6/30



your client. For example, without an evaluation table comparing actual costs for hosts, your team might select a low cost host without a proper investigation of the available software tools on the host and an unrealistically low estimate of what the client is willing to pay for a

host. The result could be lost time setting up your development environment on one host and then having to redo your work on another host that has the capacity and tools necessary for your project.

Figure 2. System Evaluation Table for Inter-Coop Council (ICC)

	Alternative 1: Open Source CRM (CiviCRM) ICC's Decision-- Priority	Alternative 2: Open Source Social Network (Elgg) ICC request – Add if easy	Alternative 3: Custom Designed System
Key Features	<ul style="list-style-type: none"> Open source Tracks donations Runs on the web Can track members, alumni, and events Automated updates Add custom fields Create contribution pages Automatically generate receipts and track thank your notes Import/export abilities View contact history Customize membership types (i.e. for each house) 	<ul style="list-style-type: none"> Alumni manage their own information Includes blogs, forums, wikis, etc. Groups for each house. Information available on ICC back end. Open source 	<ul style="list-style-type: none"> Alumni manage own info Alumni can search for other alumni and members Donor tracking system Completely customizable Automated updates
Advantages	<ul style="list-style-type: none"> Low maintenance High level donor tracking Alumni can manage own info and update No cost Can finish in 1 semester Has e-mail campaign feature Tailored to non-profits Decent automation 	<ul style="list-style-type: none"> Alumni can update and customize their info, and it is more thorough Encourages interaction User friendly 	<ul style="list-style-type: none"> Highly customizable to meet all desired requirements Reports designed for ICC High automation available
Disadvantages	<ul style="list-style-type: none"> No specific reports Relatively less customization Requires training 	<ul style="list-style-type: none"> Likely no donation tracking Not integrated with anything (property boss, QuickBooks, etc.) Not as effective for reports 	<ul style="list-style-type: none"> Difficult to create, limited time, resources, and ability Difficult to maintain Level of customization that this system allows for may be unnecessary (not worth the time and resources)
Questions			<ul style="list-style-type: none"> Does ICC really need this level of customization ?



Tips for Completeness

- Complete all the rows for each possible alternative as shown in Figures 1, 2 and 3.
- In the initial discussion with your client, always leave a blank column for the client's choice as shown in Figure 1.
- For your Project Charter, identify the client's choice, as illustrated in the header rows for ICC in Figure 2.

FAQs

Q1. What if we already know what the scope is for our project? **Answer:** Are you sure? Creating a couple alternatives that indicate different levels of functionality could be the basis of a surprising discussion. But if you really are sure, what about other choices that need to be made? Have you chosen your tools? The team for JudyPaul.com evaluated alternative scopes based on whether they needed to learn Java (the tool preferred by

their client) or stay with the Microsoft tools they learned in UT MIS classes. Their System Evaluation Table is one of the examples provided on the Resources page. What about procedures? One team in the past found it difficult to obtain a final decision on procedures for interacting with the system. Figure 3 is one of six system evaluation tables they created. Each table focused on a particular procedure like the Security Login Procedure.

Q2. What if our client wants more than we can possibly do in a semester? **Answer:** Consider illustrating your best guess of how much time would be required by your team using milestone lists like the three created for the System Evaluation Table in Figure 1. Be sure to offer alternatives that offer value to your client with clear advantages as well as disadvantages for each of your alternatives.

Figure 3. Evaluation of Alternatives for the Security Login for Survey of Organizational Excellence (SOE)

	Alternative 1: Security Login username/password text field type	Alternative 2: Security Login username/password numeric field type	Alternative 3: Security Login with session variables	Alternative 4: Security Login with cookies	Client Decision
Key Features	Set field types in Access database for username and password to text fields	Set field types in Access database for username and password to numeric fields	Use session variables within code to verify user and pass fields from page to page	Use cookies to verify user and pass fields from page to page	
Advantages	Higher security with ability for letter and number combinations	Greater support of SQL statements by JSP with Access using numeric fields	Developers have greater experience in using session variables Session variables cannot be disabled		
Disadvantages	Increased difficulty for developers in using JSP with Access and supporting certain SQL statements	Lower security with limit to numbers		Users can face hardware configuration issues—with the capability to disable cookies	



High Quality Delivery Tips

- Create a System Evaluation table for issues that are not clear decisions—this can be choice of a host, procedures, scope for the release, or any system aspect with multiple alternatives.
- Write a title that clearly identifies the issue under evaluation.
- Create meaningful alternatives to discuss with your client.
- Follow the template.
- Complete each row in the template with information that is relevant for your situation.
- Leave a blank column for the client's comments to be filled in during discussions with your client.
- For the Project Charter, clearly indicate the client's decision.

Template and Examples on Resources Page

- Evaluation Table Template with step-by-step instructions
- Evaluation Table – ICC – project scope alternatives
- Evaluation Table - 4RULost – project plan alternatives
- Evaluation Table – JudyPaul.com – development tools alternatives
- Evaluation Table – CV1 – website host alternatives
- Evaluation Table – SOE – modification log alternatives