

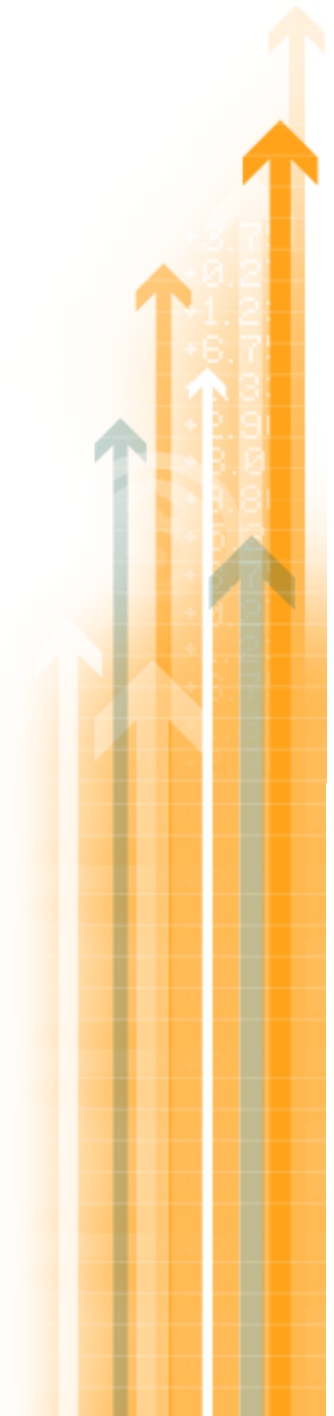
➞ Lesson Introduction

When a project nears completion, the business analyst needs to work with the developers to ensure that the final product not only functions correctly but meets the requirements originally specified (including any amendments to the requirements that may have been made along the way), and it's aligned with the organization's business objectives.

To this end, the business analyst must work with quality assurance personnel to develop testing plans and acceptance criteria based on his knowledge of what the users need and on how the users expect to use or implement the solution. These test plans should include not only functional testing but usability assessments, as well.

Next, the business analyst needs to assist in the development of a roll-out plan, which includes the following activities:

- Installing software (if applicable) or issuing instructions for a new business process (or both).
- Developing training materials or instructions to accompany the solution.
- Ensuring that ongoing business activities are not impacted by any changes:
 - Keep old software systems running while new ones are deployed.
 - Transfer data from old databases to new systems.
 - Verify that transferred data are correct and accurate.
- Developing a fall-back plan in case the new solution doesn't work as expected (of *course* it will work as expected, but it's better to have a plan in case it doesn't!).
- Developing a process for handling bugs discovered by users post-deployment.
- Developing a process for working with external actors (people or systems outside the immediate organization) to resolve issues, should they arise.



Solution Assessment and Validation

After the last stakeholder has given their formal approval for your project, it's time to get to work improving processes and building the solution. (In "real" projects, some of the groundwork associated with the "build" phase may actually have been carried out before the last signature appeared on the approval documents. For purposes of this discussion, however, we'll assume everything proceeds in "proper fashion" and that work begins only after the project has been fully and officially approved.)

The business analyst may play a variety of roles in the building phase of a project depending on the organization and how development work is customarily assigned. Ideally, the business analyst should stay intimately connected to the project throughout the building (which includes implementing process improvements), acceptance testing, and deployment activities. After all, the business analyst may have the best understanding of the project requirements and end-user needs. In smaller organizations, the business analyst may actually carry out some of the development tasks while in larger organizations she may provide guidance and oversight to those who carry out the work.

Note: The Project Manager will be directly responsible for coordinating the business analysis activities in this phase of the project, for example in allocating requirements to the solution while managing the project scope, costs and budget. It's important to understand the role of the Project Manager in each one of the tasks in this Knowledge Area. For more detail, see the **Stakeholders** section in the BABOK® for all the tasks in Chapter 7.

➞ Lesson Objectives

Upon completing this lesson, you should be able to:

- Describe the role of the business analyst in the development, acceptance, and deployment phases of a project.
- Understand the importance to the organization in assessing a new solution.
- Define the allocation of requirements to solution components.
- Describe the activities that are involved in assessing the readiness of an organization to make effective use of a new solution.
- Define *Transition Requirements* and their role in implementing a new solution.
- Describe the activities and techniques used to validate whether a solution meets the needs of the organization, and also to respond to identified defects.
- Describe the activities and metrics used to evaluate the performance of a functioning solution after it's been implemented.

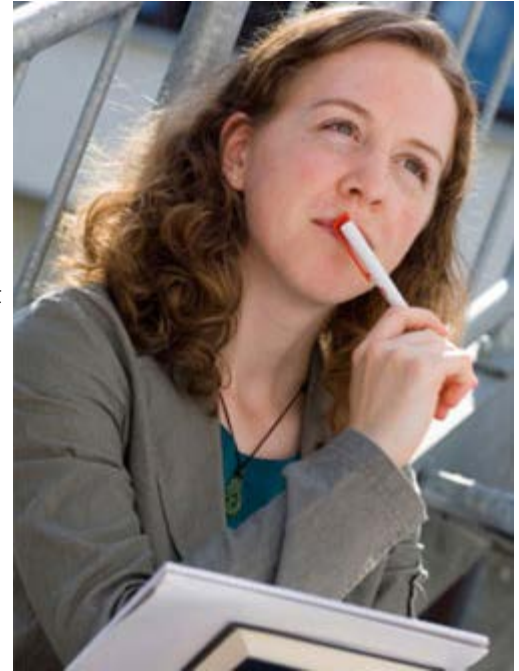


Assess the Proposed Solution

Before a solution is actually implemented, the business analyst needs to review the proposed solution to determine how closely it matches what the stakeholders actually asked for.

In assessing the proposed solution, the business analyst reviews the requirements and how they've been prioritized to confirm that the most valuable requirements are met (ensuring that, by meeting key requirements, business objectives will be achieved) for the good of the organization and the stakeholders.

Solution assessment also entertains multiple proposed solutions. When you have to deal with more than one solution, it becomes even more difficult to assess the requirements within each solution and how the solutions compare with one another in terms of value for the stakeholders.



Solution Options

One of the key questions when developing a solution strategy involving technology is whether to buy an off-the-shelf product or design one in-house. Business analysts can assist technical personnel in determining how well each option under consideration will meet the stakeholders' requirements.

Additionally, business analysts can help make decisions about specific technologies from among a set of competing options.

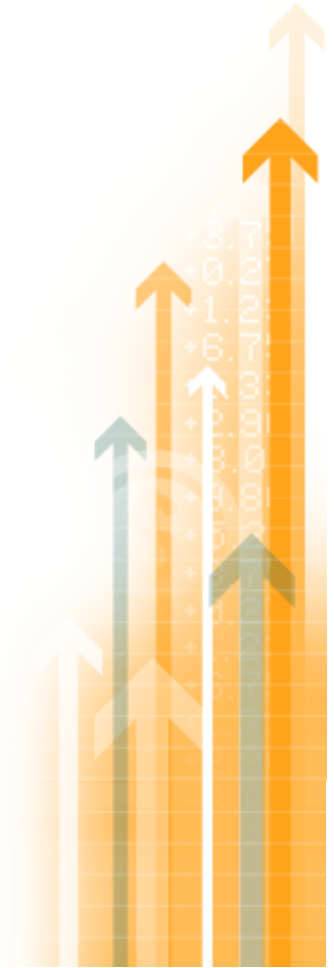
Once the designers have proposed several solutions, the business analyst often leads the process of choosing one of them, weighing multiple factors including whether it meets the requirements or is feasible.

Solution options will sometimes offer capabilities - either actual or potential - beyond those identified for the current solution. These capabilities might not be of immediate value to the organization now, but have the potential to provide future value. For example, in the case of a software solution, these additional capabilities could be implemented as new features in the next release of this software.



➡ Self Check

1. **The technical team working on a database implementation project recommends using a very low-end database server. The team members have also insisted that they don't need much information to solve the problem. In their minds, the solution is obvious. Why should you, as the business analyst, insist on exploring alternative solutions for the project customers?**
 - a. You should explore other solutions to see if a more cost-effective database server is available.
 - b. You should explore other solutions to see which one fits the customer requirements the best.
 - c. There's really no need to look at other solutions since the technical team has already identified the one that is most cost-effective.
 - d. Additional solutions are needed so that the project customers can select the solution they like the most.



Allocate Requirements

The *Allocate Requirements* task describes the activities and techniques used to allocate *stakeholder* and solution requirements among solution components in order to maximize the business value of the solution to the stakeholders.

This task is the process of determining **which components of a solution** should be implemented in what release or version of the project deliverables. This allocation is important because the business value of a solution can change depending on how requirements are implemented and when the solution is actually available for use.

Note: Requirements allocation must follow requirements prioritization, which should have been done during elicitation, for example in a *Requirements Workshop*, or also based upon the business case's key requirements.

Develop Solution Components

Once the requirements are fully defined and documented, designers and developers can begin crafting the solution. Each solution component implements a subset of these requirements. Because there are likely to be numerous ways to achieve the same functionality, solution developers must begin by considering multiple alternatives. The allocation of requirements to solution components will be a primary driver of the cost to implement the solution and the benefits delivered by it.

Here are just a few simple examples from the world of software application development:

- Which operating system: Microsoft or an open-source system?
- Which database will we use on the back-end: Microsoft SQL Server, Oracle, or an open-source database (such as MySQL)?
- Which programming environment is better for this application: .NET or Java?
- Should we build the entire project in-house or should we outsource some (or all) of it?
- Do we want Windows applications or web-based applications? (That is, should the software applications run only in a Microsoft Windows environment or should they run in a web browser and be accessible from any computer with an Internet connection)?
- Which algorithm is best (or most efficient) for this application?
- Will a software or IT solution really solve the problem or is it more cost-effective to modify (i.e. improve) the business process first?

From Requirements to Design

Functions and Features

A technical solution must be able to "do something." We can summarize its capabilities in terms of *functions* and *features*. A function is a specific action that a solution can "make happen," like opening a subdirectory or connecting a telephone call. A feature is usually a combination of functions that logically belong together.

For example, most software packages have features that allow users to save their documents. The "save" feature encompasses several functions such as changing the name of a document, converting it to a different format, moving it to different directories, creating a new subdirectory in which to store it, and so forth - all included as part of the fundamental, high-level "save" instruction.

The business analyst must map (or link) specific requirements to specific functions and features that will be incorporated in the solution by reviewing documentation developed during the requirements analysis phase to ensure that all requirements eventually appear in the solution as functions or features. Very often, this is done somewhat formally using spreadsheets (or requirement databases) to list all requirements and the types of functionality that will be needed to implement them. During this process, designers need to be aware of any constraints (from any source) that may make it impossible to meet a particular requirement. The business analyst plays a vital role here in ensuring that key requirements are not compromised.

Naturally, the testing and verification of the completed solution begins by exercising each function to ensure the proper outcomes. When you click on "print," does the document actually appear on my printer? Is it formatted correctly? Are all the functions within the "print" feature working correctly? Even if the technology is not an IT solution, you still need to test its functions in such a systematic manner. Of course, the tester must keep careful records of both the test and the results so that the solution's developers can fix the problems. Here is where a business analyst can provide guidance and oversight.

Release Planning

Building the solution involves a series of steps that can be divided into distinct phases. The business analyst works with stakeholders and the technical design team to determine the number of design phases that will be necessary, taking into consideration the project budget, the order in which certain features need to be deployed, resource constraints, deployment constraints, the amount of training that will be needed , and the availability of users to undergo that training.

For example, suppose new features are being planned for a university student enrollment management system that will:

- Store student data (name, address, etc.).
- Store student course histories and grade transcripts.
- Allow students to enroll in courses online (over the web).
- Allow students to drop courses online (over the web).
- Allow students to print unofficial transcripts.

The dean's office has mandated that the ability to store all student data - including grade transcripts - and to enroll in courses online have the highest priorities. The other functions can be handled manually, as in the past, for the time being.

The business analyst on the project might decide to break this upgrade project into three major phases:

- **Phase 1:** Add storage and online enrollment capabilities.
- **Phase 2:** Add course drop capability.
- **Phase 3:** Add transcript printing capability.

Phases 1 and 2 have approximately equal priority, and Phase 3 can be completed later, as time and resources permit.

Assess Organizational Readiness

As the new solution is being designed, the business analyst needs to assess:

- the effect this solution will have on the organization
- whether the organization is prepared for the impact the implementation will cause

In order to identify and gauge the solution impact, the business analyst needs to understand what changes will occur in the business, technical or operational areas of the organization, and whether these changes will affect other business units or areas of operations.

Effective (planning for, and) communication of the solution impacts can assist in mitigating some of these changes and can set in motion the necessary organizational change management processes. Assessing the organization readiness can also help identify additional requirements (i.e., training) for solution implementation. These types of requirements are referred to as *Transition Requirements*.



Solution Impact Assessment

The impact of the solution can be assessed from different view points. For example, culturally, the beliefs, attitudes and feelings of key stakeholder groups are assessed, as well as their willingness to accept change. Another assessment can be about a user's group **technical** skills and whether training will be necessary for them to use the solution effectively.

A technique that can be used to assess the level of support or resistance to a change is *Force Field Analysis*.

Figure 7-5: Force Field Analysis Diagram



This graphical technique places the forces for and against a change (and their relative strength) on opposite sides of a line in order to assess which side is stronger. [Here](#) is more information about this technique.

Figure 7-5: Force Field Analysis Diagram, Business Analysis Body of Knowledge® (BABOK® guide), Version 2.0, 2009, International Institute of Business Analysis, Toronto, Ontario, Canada, page 130.

Users

When beginning to create a solution, the first step is to take a look at the future users of the solution. The solution must be tailored to "fit" those users in terms of a set of user-focused dimensions while meeting the functional specifications that were determined during the requirements elicitation process.

One important element deals with security. Not all users need to access all the data on the system.

For instance, consider an organization's human resources system. We expect that only HR employees would need to view employee data such as social security numbers and salary figures. Consequently, only HR employees would be granted access to the HR database and to any applications that make use of HR data. Further, of those employees, only a few would have rights to *modify* or *delete* data.

Now, we may want the same system to allow all employees to go online and update their own addresses and other non-sensitive personal data. So, you would design web accessibility for less-sensitive data (password-protected, of course) while allowing only authorized employees to access more sensitive information, perhaps from computers or terminals within the organization's human resources department as an added layer of security.

Technology Proficiency

Another consideration is the users' degree of proficiency with technology. Some users are very comfortable with multi-featured user interfaces and can figure out how to coax the application into doing what they want. Others are more computer-shy and need simpler interfaces with more explicit instructions.

We might expect our organization's employees to be more computer-savvy but cannot make this assumption about members of the general public. A web application to be used by an organization's customers obviously needs to have more help files and error-trapping capabilities than an application to be used by information services staff members within that organization.

Define Transition Requirements

Transition Requirements are "a classification of requirements that describe capabilities that the solution must have in order to facilitate transition from the current state of the enterprise to the desired future state, but that will not be needed once that transition is complete." (BABOK® Glossary, page 234).

Typically, a solution is designed to replace an existing (deployed) solution. In many cases, there is a transition period when both new and existing solutions run in parallel. To complete the transition from the existing system to the new system, the new system needs to get information from the old system (i.e., data, files, etc), users need to be trained on how to operate the new system, and so forth. There might even be specific activities that need to happen (i.e., data conversion) before the new system is fully operational. These one-time requirements that are only needed to complete the transition to the new system are *Transition Requirements*.

Keep in mind that except for their unique nature, transition requirements are elicited, analyzed, managed and communicated by performing the same tasks as for any other requirements.

Transition Requirements Considerations

Examine both the solution currently in place and the proposed new solution to identify the following:

- Do the current features behave substantially different from the new solution?
- Is there any information that needs to be transferred to the new solution?
- Is there a need for data conversion?
- Is any current information going to be archived?
- How is the ongoing work going to be managed as the organization moves to the new solution?
- Are the users of the new solution trained to use it?

The move to a new solution can bring significant organizational change. The business analyst could help develop recommendations for changes to the organizational structure or personnel, as job functions may change significantly as the result of work being automated.

Validate Solution

Solution validation is required to ensure that a delivered solution meets the business needs on an ongoing basis. Note that validation can only be done *after* the new solution is implemented.

The main goal with solution validation is two-fold:

1. Ensure that the delivered solution meets the stakeholder needs as defined in the validated and prioritized stakeholder and solution requirements.
2. Determine the most appropriate action for identified defects (i.e., failures to meet a stakeholder need).

Knowing the requirements, the business analyst knows what needs to work and how it should work. She can help QA personnel and testers identify critical functionality for testing.

Note also that a technology-based solution may operate correctly and meet all functional requirements. However, if it is difficult to use operators may make errors or may not use the system's entire range of capabilities. The business analyst helps designers see how to make their products more usable and, subsequently, more acceptable to stakeholders and end users.

Acceptance Testing

People (or, collectively, departments in an organization) who commission the development of a software application must specify **acceptance criteria** that the software must meet in order to accept delivery and implementation. Acceptance criteria are a subset of the user requirements that are absolutely necessary for delivery to take place. Applications that fail one or more **acceptance tests** are sent back to the developers for rework.

Here is where the business analyst plays a key role. The analyst knows intimately the complete set of user requirements and can provide a link between the end users and the developers by documenting the acceptance criteria and assisting with the creation of an acceptance testing plan.

In most organizations business analysts do not carry out the testing themselves, but they are involved in reviewing the results and in discussing those results with their "clients," the end users. They can then communicate any problems that may have arisen to the technical people on the development team.

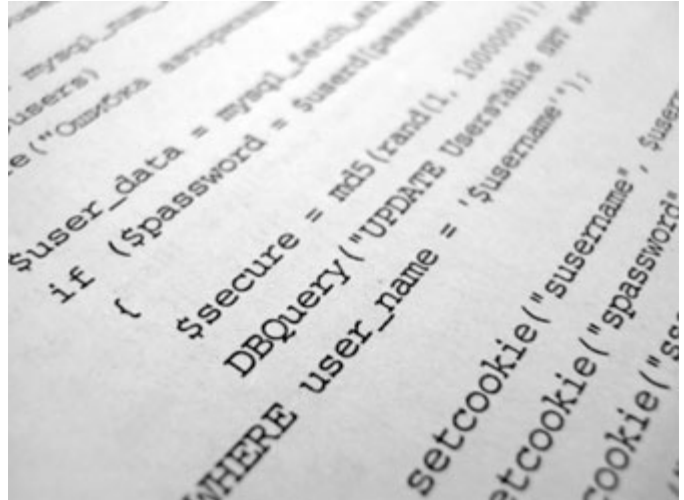


Software Testing

Ideally, the ability to test and validate a software solution should be built into the design from the very beginning. This allows developers and business analysts to catch and correct potential problems while they are still small. If a problem with the logical design or functionality appears early in the project, it is much easier and much less costly to fix it then than to wait until the product is nearly ready for delivery to end users.

Consider, for example, a web-based application for conducting e-commerce. Suppose that the application architect decides to use a relational database for storing all data pertaining to the e-commerce functions including not just product and customer data, but working data that are used in a scratch-pad fashion as well. A developer might create a database interface routine that is used in hundreds or thousands of places in the application - wherever data needs to be stored or retrieved. If a problem arises with this routine, nearly the entire application will be affected.

It's better to ensure that the routine works correctly at the beginning of the development process while there are relatively few modules affected, than at the end of the process when major design decisions may have been made based on this routine. Typically, however, the testing plan would encompass interface testing among components early in the project. These "component interface requirements" will be tested and validated to ensure the components meet these requirements.



Not an Exact Science

As an application increases in size, complexity, and functionality, the number of possible sequences of events increases exponentially - each of which could be a prospective testing pathway if one wanted to test absolutely every possible combination of button presses, mouse button clicks, etc.

In reality, it is neither feasible nor possible to test absolutely everything, but quality assurance professionals have devised ways to test the critical functions and reduce the greatest number of potential failure modes in the most economical manner. It is much easier to assess quality in terms of how the application was put together or validity in terms of whether the application does what people want it to do.

Of course, certain software applications need more rigorous testing than others - the software that controls your airliner's autopilot system deserves more scrutiny than your word processing application, for instance.

Software Testing

For manual and even for some automated testing of a given application, a test engineer begins by creating a test plan, which consists of all the functions and features that the application must provide to implement a set of business rules. Each business rule is then accompanied by a test transaction designed to determine whether the business rule is implemented correctly.

A common testing methodology used today involves releasing early versions ("beta versions") of software packages to a limited number of end users and allowing them to report "bugs" back to the manufacturer. Thousands of amateur testers are likely to exercise nearly all of an application's functionality and, because they do their work in parallel, dramatically decrease the amount of time needed to cover a broad swath of the application's features.

No software is ever released completely bug-free. Look at how Microsoft periodically issues updates and service packs that address issues that are uncovered months or years after a product is released.

Releasing beta versions is most useful for "shrink-wrapped" applications, that is, for products that are to be released to the general public. In the case of a specialized, in-house application that will be used only by a few organization staff members, releasing beta versions may not be especially useful, however.

Once developers obtain a good list of defects, they can begin addressing those defects systematically to "fix" problems with coding, the interpretation of user requirements, problems with the test data, etc. that gave rise to the unexpected performance the testers observed. Testing and fixing bugs may involve several iterative cycles with a product being officially released when the cost of additional testing (in terms of lost income, lost market position, the salaries of the testers, and other factors) exceeds the costs that may accrue from the release of a faulty product (in terms of lost reputation, products returned for a refund, and even lawsuits when critical functions are mishandled by an application).

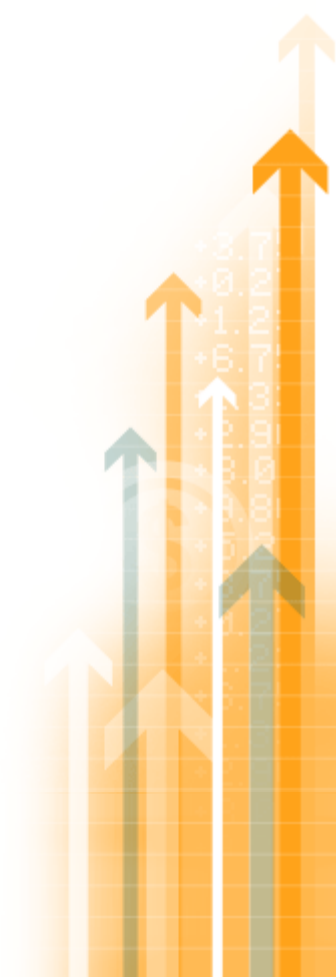
Evaluate Solution Performance

Solution evaluation (also called post-implementation assessment) looks at the deployed solution to see how it is being used and also gauges any positive/negative effect on the organization. Also, it aims to describe how the solution is performing in relation to business goals and objectives.

The performance of the solution will be measured against the business requirements and must consider any known defects in assessing the quality of the solution. There should also be a set criteria (Solution Performance Metrics) by which the performance of the solution is to be assessed. These criteria can be quantitative (number hours of constant operation), qualitative (customer satisfaction) or even defined in other measures unique to the project stakeholders (i.e., opinions).

It is always important to evaluate lessons learned so the next project goes more smoothly. A thorough review can identify rough spots from stakeholders' and solution developers' perspectives. This information is then applied to the next project.

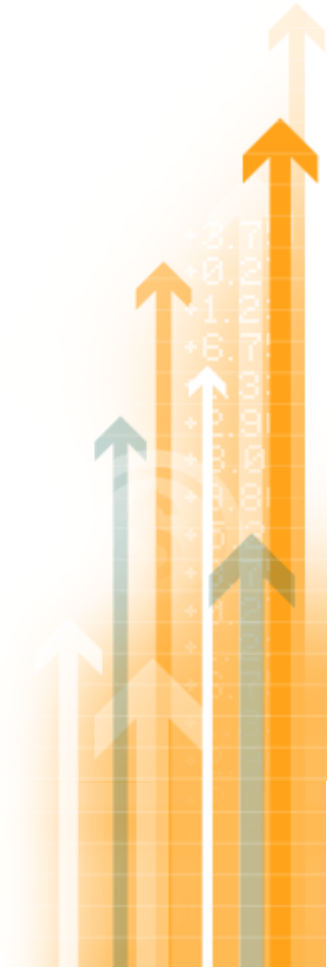
➡ Self Check



➔ Self Check

1. What are the main goals of solution validation? Choose all that apply.

- a. You should explore other solutions to see if a more cost-effective database server is available.
- b. To validate that the organization is ready for the business changes brought on by the solution
- c. To determine which components of a solution should be implemented in what version of the project
- d. To determine the most appropriate action for defects

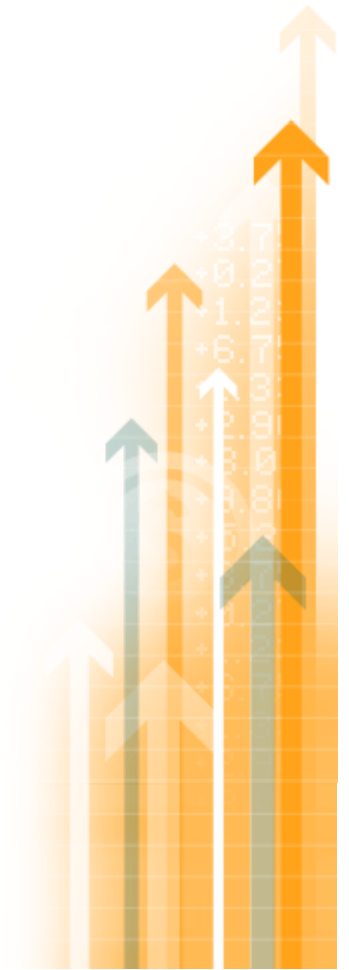


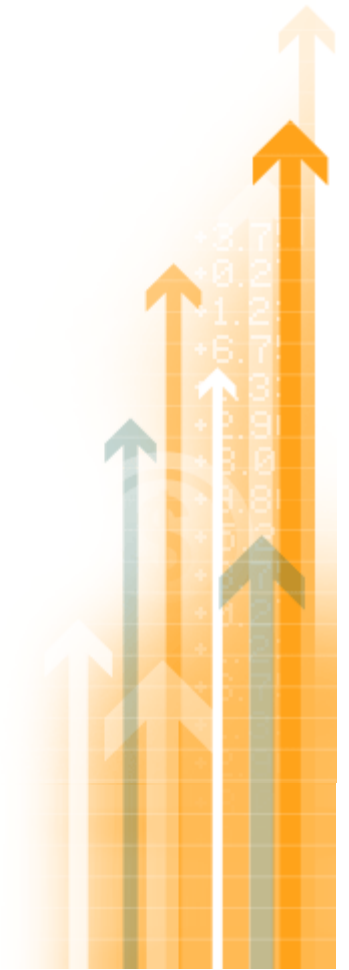
➡ Self Check

2. Because no software is ever free of defects, acceptance criteria are typically optional on most software solution projects.

True

False





➔ Lesson Summary

In this lesson, you learned about the *Solution Assessment and Validation* Knowledge Area. Key topics included:

- Assess Proposed Solution
- Allocate Requirements
- Assess Organizational Readiness
- Define Transition Requirements
- Validate Solution
- Evaluate Solution Performance

The *Solution Assessment and Validation* Knowledge Area describes the tasks that are performed in order to ensure that solutions meet the business need and to facilitate their successful implementation.

