The IBM Rational Unified Process for COTS-based projects: An introduction

Cécile Péraire
Russell Pannone

August 15, 2005

Building solutions based on pre-existing Commercial-Off-the-Shelf (COTS) packages is different from typical green-field development. Numerous projects have unsuccessfully tried to integrate pre-existing COTS packages by defining the requirements, formulating an architecture to meet those requirements, and then trying to fit COTS packages into that architecture. The unique characteristics of COTS packages introduce dynamics and specific constraints that must be accommodated. Projects that build solutions based on COTS packages require dedicated guidance.

This paper will introduce the IBM Rational Unified Process® for Commercial-Off-the-Shelf Package Delivery (or RUP® for COTS). The RUP for COTS is a configuration of the IBM Rational Unified Process, or RUP, that addresses the needs of IBM customers who demand process guidance in their projects in order to evaluate, recommend, acquire, install, configure, field, and evolve solutions based on COTS packages. In this paper, we will first introduce some best practices for COTS projects. Then, we will define the foundations of the RUP for COTS. Finally, we will walk through each RUP phase of a COTS-based project by introducing the objectives, roadmap, roles, activities, artifacts, and milestone of the phase.

The RUP for COTS is based on the Evolutionary Process for Integrating COTS-Based Systems (EPIC), which is a proven methodology developed by CMU-SEI (Carnegie Mellon University-Software Engineering Institute). Specifically, the RUP for COTS is based on EPIC concepts and principles (spheres of influence, iteratively converging decisions, accumulating knowledge, and increasing stakeholder buy-in). In addition, the basis of some phase objectives, roadmap descriptions, artifacts, and milestones come from EPIC.
Best practices for COTS projects

The goal of this section is to identify the unique characteristics of COTS packages and derive some best practices for COTS projects from these characteristics. We will start by considering the definition of a COTS package.

A COTS package

is a software product that is:

• Supplied by a vendor
• Sold, leased, or licensed to an acquirer
• Used without modification of its code
• That may be tailored to be integrated into the acquirer environment
• Supported and evolved by the vendor, who retains the intellectual property rights

Based on this definition, we can identify a list of unique characteristics of COTS packages:

• The marketplace, not the needs of a single system, drives COTS package development and evolution.
• COTS packages and the marketplace undergo frequent, almost continuous, change.
• Frequency and context of COTS package releases are determined at the discretion of the vendor.
• COTS packages are built based on unique architectural assumptions that may not be applicable to the target organization.
• At best, consumers have limited visibility into COTS package internals and behavior.
• COTS package assumptions about end-user processes may not match those of a specific organization.
• A COTS vendor is not a subcontractor. Acquirers seeking to influence package changes require a different type of relationship to the software supplier.
• COTS package components often have unsuspected dependencies on other COTS package components.

In order to take these unique characteristics of COTS packages into account, any process that builds, fields, and supports solutions based on COTS packages must support the following best practices:

• Create an environment where COTS packages (and therefore the marketplace) drive the evolving definition of the solution. Since the control of COTS packages is in the hands of the COTS package vendors, and because COTS packages undergo frequent changes, an environment that facilitates hands-on analysis of the COTS packages and continuous negotiation with stakeholders will be required for the life of the project. This will allow the evaluation of new and changed COTS packages and their impacts on evolving solutions.
This will ensure that the marketplace drives the solution definition as much as other major influencing factors (such as the business and stakeholder needs).

- **Compose solutions from a diverse range of components.** Solutions are defined by combining pre-existing components (such as COTS package components). Insight into the inner workings of these components will vary (e.g., black, white, and gray box) depending on the source and the intended use of the component. Therefore engineers may have to infer the behaviors of various component combinations as they integrate these components.

- **Implement disciplined iterative and risk-driven systems engineering practices.** An iterative and risk-driven definition of COTS-based solutions allows identifying and addressing high project risks early in the project, when major changes remain possible and are not too expensive. As an example, a risk could be the end-user resistance to embracing the changes to end-user business processes inherent in a COTS package. Iterative development encourages frequent and direct feedback from all the affected stakeholders through evolving prototypes that characterize and mature the architecture while reducing risk in the solution.

- **Support concurrent and integrated implementation of engineering, business, and procurement activities.** An engineering decision to include a new COTS package is also a decision to modify some of the organization's business processes according to the ones supported by the package, as well as a decision to acquire the package from its vendor. Therefore, the engineering, business, and procurement activities of the project must be coordinated to support the flexibility and negotiation of requirements and iterative definition of the solution.

- **Balance component obsolescence and solution stability.** Due to the volatility of the marketplace, a new COTS package or a new release of a COTS package being used can be introduced at any time during building, fielding, or support of the solution. However, the need for upgrading or introducing new COTS packages must be balanced with the need for providing a stable solution to the users.

- **Develop and maintain a resilient system architecture as a centrally managed asset.** Since the COTS packages are "owned" by the vendors, the framework by which the components are linked to support the organization's needs -- the architecture -- becomes a key asset within the organization. A resilient architecture that can retain its structure and cohesiveness while allowing the system to respond easily to changes becomes an important strategic asset to an organization.

- **Incorporate activities for changing the end-user business processes.** Using COTS packages in solutions does not simply imply automating a predefined set of business processes. Since COTS packages embody the vendor view of end-user business processes, changes to the end-user business processes must be negotiated based on those processes that the COTS packages under consideration were designed for. Engineering activities should be coordinated with activities for changing the end-user business processes. The definition and implementation of end-user business process and organizational changes must be coordinated through the life of the project.

- **Maintain and document how COTS packages support the solution.** The ways in which COTS packages are actually used within the organization must be tracked and captured. This information is important to the evaluation of new and changed COTS packages and their impacts on evolving solutions. Many COTS packages have diverse functionality, not all of which is required in a given solution. It is important to document the functionality applicable to
the solution. It is just as important to capture how any "excess" functionality is handled within the solution -- especially any functionality that is blocked or bypassed -- to be able to ensure that none of these "excess" capabilities will find their way into operational use after fielding.

The RUP for COTS is based on these best practices. The RUP for COTS provides IBM customers with process guidance in order to evaluate, recommend, acquire, install, configure, field, and evolve COTS package solutions.

A COTS package solution is the integrated assembly of one or more of the following:

- COTS packages or COTS package components
- Legacy systems (piece of the systems being replaced)
- Reuse libraries and other reuse sources (e.g., freeware, shareware)
- Any required custom code (including wrappers and "glue")
- Appropriate linkage to the broader organization’s architecture with which the solution must interface
- Changes to the end-user business processes necessary to match the processes provided in the COTS packages

Foundations of the RUP for COTS plug-in

The following sections introduce the EPIC concepts and principles that form the foundations of the RUP for COTS.

The four spheres of influence

The four spheres of influence represent competing interests that must be considered in forming a viable solution that effectively leverages preexisting COTS packages from the marketplace. The four spheres of influence are:

- **Sphere 1 - Stakeholder needs and business processes**: This sphere denotes requirements (including quality attributes, such as performance, security, and reliability), end-user business processes, business drivers, and operational environment.
- **Sphere 2 - Architecture and design**: This sphere denotes the essential elements of the system, their relationships, and how they fit within the enterprise system. The elements include structure, behavior, usage, functionality, performance, resilience, reuse, comprehensibility, economic and technologic constraints, and tradeoffs.
- **Sphere 3 - Marketplace**: This sphere denotes available and emerging COTS technology and products, non-development items, and relevant standards.
- **Sphere 4 - Programmatic and risk**: This sphere denotes the management aspects of the project. These aspects consider the cost, schedule, and risk of building, fielding, and supporting the solution. Key to these management aspects are the cost, schedule, and risk of changing the necessary business processes.

These four spheres are simultaneously defined and traded through the life of the project because a decision in one sphere will inform and likely constrain the decisions that can be made in another sphere. For example, a stakeholder need may be stated in such a way that it cannot be satisfied by any known pre-existing COTS package.
Iteratively converging decisions
In order to maintain balance between the four spheres, EPIC creates an environment that supports the iterative definition of the four spheres over time while systematically reducing the trade space within each. This allows a decision in one sphere to influence, and be influenced by, decisions in the other spheres. Initially, as shown to the left in Figure 1, the trade space may be large. There is flexibility for making tradeoffs between the stakeholder needs and end-user business processes, the architecture and design, the offerings of the marketplace and other sources, and programmatic and risk. As EPIC is used to drive toward a refined understanding of the solution, the tradeoff space shrinks. The spheres increasingly overlap as fewer decisions remain in any single sphere that can be made without significant impact on the others.

Figure 1: As knowledge and stakeholder buy-in grow, the four spheres of influence increasingly overlap.

An iterative development process is necessary to keep the requirements and architecture fluid as the four spheres of influence are considered and adjusted in order to optimize the use of available COTS packages. Each iteration contains activities that gather information from each of the four spheres. Each iteration refines the newly gathered information through analysis and negotiation with affected stakeholders to form the harmonized knowledge needed to assemble an executing system implementing the solution and supporting the needed end-user business processes. The iterations are managed by the four RUP phases and associated milestones, as shown in Figure 2.

Figure 2: The four spheres of influence in the context of RUP phases and iterations

Accumulating knowledge
Concurrent with diminishing the tradeoff space between the four spheres of influence, knowledge about the solution must grow at a controlled pace. This knowledge is reflected in the set of artifacts
necessary to evaluate, recommend, acquire, install, configure, field, and evolve the solution. Most of the artifacts are started in outline form and are expanded as more information is gathered and refined. This knowledge includes an increasingly detailed understanding of the following:

- Capabilities and limitations of candidate COTS packages, the vendors that produce them, and the marketplace drivers that control them
- Negotiated and prioritized stakeholder needs and end-user business processes
- Architectural alternatives and mechanisms to integrate the COTS packages into the acquiring environment
- Implications of the COTS packages on the stakeholder needs and end-user business
- Planning necessary to implement and field the solution (including any needed end-user business changes) and associated cost, schedule, and risk

It is particularly important to keep knowledge current about the COTS packages critical to the solution and the marketplace or other sources that supply COTS packages. This allows the organization to track trends that may affect the solution over time. This also allows them to keep the volatility of the marketplace in balance with the need for stability in building, fielding, and supporting the solution in operations. Monitoring and evaluation begin at project initiation and continue until the solution is retired.

**Increasing stakeholder buy-in**

While decisions are converging and knowledge is accumulating, the stakeholders must increase commitment to the evolving definition of the solution. Since these stakeholders may be a broad and possibly disparate group, this will be difficult for many projects because this is a significant commitment and may be unprecedented within the organization. Active participation from the stakeholders, however, is essential to the success of the solution. Creating an environment that includes the stakeholders (or empowered representatives) directly affected by any change in end-user business processes allows a quick resolution to discovered mismatches among elements, such as the available COTS packages, the desired end-user business processes, and the stated stakeholder needs. Just as important, a stakeholder-friendly environment can demonstrate that the solution can be built within cost and schedule constraints with acceptable risk.

End-user needs mature and change with increased understanding of available COTS packages. The day-to-day involvement of end-users is essential because the activities that identify, evaluate, and select COTS packages will shape the end-user business processes and define the functionality that will be delivered. At the same time, architectural stakeholders ensure that the COTS packages considered can be effectively integrated within the broader organization's existing systems to meet required performance parameters. Business analysts must ensure that viable vendors support and evolve the COTS packages. Vendor involvement can provide enhanced visibility into the COTS package's capabilities and potential insight for the vendor into the organization's needs. The continuous negotiation and reconciliation among affected stakeholders leads to a more effective use of COTS packages in satisfying the mission.

The stakeholders confirm and increase their buy-in and commitment to the evolving definition of the solution based on the iterative and incremental growth of functionality that is continually
demonstrable. An executing system -- not merely plans and forecasts -- is essential to reduce risks due to misunderstandings or unforeseen technical and operational factors.

How are the RUP for COTS project phases defined?
This section walks through each of the four RUP phases (Inception, Elaboration, Construction, Transition) of a COTS-based project. It introduces the objectives, roadmap, roles, activities, artifacts, and milestone of each phase.

The Inception phase
The goal of the Inception phase is to achieve concurrence among all stakeholders on the lifecycle objectives for the project. The Inception phase establishes that the project is worth doing and that one or more feasible candidate solutions exist. The following sections describe the Inception objectives, roadmap, roles, activities, artifacts, and milestone.

Inception objectives
In the context of a project that performs COTS package evaluation, recommendation, acquisition, installation, configuration, fielding, and evolution, the primary objectives of the Inception phase include:

1. Establishing the project's software scope and boundary conditions, including an operational vision, acceptance criteria, and what should be included in or excluded from the solution.
2. Identifying the stakeholder needs and desired business processes, and discriminating the critical use cases and non-functional requirements of the solution (the primary requirements that will drive the major tradeoffs).
3. Determining the architectural and design constraints imposed on the solution by any infrastructure on which the solution will run and any other systems with which the solution must interact.
4. Estimating the project management constraints (time, money, people, etc.) for the project, potential risks, as well as tolerance for and inhibitors to implementing changes across the organization.
5. Identifying the offerings of the marketplace in terms of relevant COTS packages and vendors.
6. Defining candidate solutions that could meet the demands of the most critical use cases and non-functional requirements, architectural and design constraints, and project management constraints with reasonable risks. A candidate solution is typically the integrated assembly of one or more COTS packages or COTS package components, legacy systems (piece of the systems being replaced), reuse libraries, other reuse sources (e.g., freeware, shareware), any required custom code (including wrappers and "glue"), appropriate linkage to the broader organization's architecture with which the solution must interface, and changes to the end-user business processes necessary to match the processes provided in the COTS packages. Note that some candidate solutions may not be based on COTS packages at all, but the assumption made in this RUP configuration is that at least one candidate solution will be based on COTS package(s). Defining candidate solutions requires exposing mismatches and negotiating tradeoffs among the critical use cases and non-functional requirements, architectural and design constraints, project management constraints, and risks. Note the importance of understanding the impact of these tradeoffs on end-user business processes.
7. Demonstrating, for each candidate solution, the feasibility of one or more candidate architectures against some of the architecturally significant requirements, including critical use cases and non-functional requirements. The demonstration is done by assembling an architectural proof-of-concept that may take many forms, like a sketch of a conceptual model of the solution, a simulation of a solution, an executable prototype, or a combination of these different elements.


9. Preparing the supporting environment for the project by setting up the experimentation facility. The experimentation facility replicates, as closely as possible or practical, the operational environment (including interfaces to any important external systems).

Most of these objectives are applicable to any RUP software development project, except for Objectives 5 through 8, which are fairly specific to COTS package projects.

Inception roadmap

The focus of the Inception phase is first on gathering information from each of the four spheres of influence and getting concurrence among the stakeholders around that information.

Once the information is gathered from the spheres, it is combined and analyzed to define candidate solutions that could meet the demands of the most critical use cases and non-functional requirements, architectural and design constraints, and project management constraints (time, money, people, scope, etc.) with reasonable risks. This is done by performing a gap analysis between the project needs and what is available on the market. The goal is to understand where they match, where they do not match, and to negotiate tradeoffs with relevant stakeholders in case of mismatches. Based on this understanding, candidate solutions are formed, and a draft of the architecture is outlined for each solution.

The feasibility of each candidate solution should be demonstrated by assembling an architectural proof-of-concept. An architectural proof-of-concept may take many forms, such as a sketch of a conceptual model of the solution (using a notation, such as UML together with some screen shots), a simulation of a solution, an executable prototype, or a combination of these different elements. The architectural proof-of-concept helps demonstrate the candidate solution feasibility from a technical point of view, but also from a business point of view. Indeed, the architectural proof-of-concept should be used to prototype the changes to end-user business processes associated with each solution.

Dealing with business changes is probably the most challenging part of a COTS project, so it's important to involve the stakeholders affected by the solution early in the process, and make sure they really understand these business changes and agree to implement them should the solution be selected.

The definition of candidate solutions and assembling of architectural proofs-of-concept allow identifying the most relevant feasible candidate solutions that meet the demands of the critical use cases and non-functional requirements, architectural and design constraints, and project
management constraints with reasonable risks. A recommendation, based on business goals, is then made to further investigate these candidate solutions in later iterations.

Note that, in the Inception phase, the goal is not to gather detailed information from each sphere, nor to define candidate solutions in detail. Instead, the objective is to focus only on the critical information and aspects of the candidate solutions necessary to achieve the goal of the current iteration. Figure 3 illustrates typical iterations in the Inception phase.

**Figure 3: Typical iteration in the RUP Inception phase of a COTS-based project**

There is at least one iteration in the Inception phase. The number of further iterations required depends on the complexity of the candidate solutions and on the magnitude of mismatch between the initial expectations of the stakeholders and the capability that can be fielded with reasonable cost, schedule, and risk.

**Essential roles, activities, and artifacts in Inception**

This section presents some examples of roles, activities, and artifacts that are new to RUP, which were introduced because they are specific to COTS-based projects. These elements play a significant part in the Inception phase, even though they may be involved throughout the project lifecycle. Refer to the Appendix for the complete list of new elements.

The new **Acquirer** role plays a key part in the Inception phase. The Acquirer oversees the evaluation, recommendation, and acquisition of COTS packages. In Inception, the Acquirer is responsible for performing a new activity called **Identify Relevant COTS Packages and Vendors**. One objective of this activity is to capture the broad characteristics of the market represented by a set of competing COTS packages under consideration for use in the solution. These characteristics include vendors and acquirers participating in the market, COTS packages offered, processes automated, technologies represented, procurement strategies practiced, and competitive market forces. The focus is on large-scale market dynamics rather than in-depth analysis of individual COTS packages. Another objective is to identify the COTS packages that may be applicable to the solution. The task that leads to this information is commonly termed "product evaluation." It entails the examination of COTS package features and supported business processes, and we recommend doing this as much as possible through product experimentation. The information gathered in the context of the new Identify Relevant COTS Packages and Vendors activity is captured in the following new artifacts: **Market Segment Information**, **COTS Package**
**Dossiers**, and **Vendor Dossiers** (one dossier per relevant COTS package and one dossier per vendor, so we avoid redundancy when one vendor sells more than one COTS package, for instance).

In addition, the process provides the flexibility to decide whether or not to use some more formal techniques for product evaluation, like inviting the vendors to respond to a Request for Information (RFI), a Request for Proposal (RFP), or to demonstrate their products in an evaluation workshop. The process allows various levels of formality depending on the needs and practices of the acquiring organization.

Figure 4 illustrates some of the new roles, activities, and artifacts that play an important part in the Inception phase.

**Figure 4: Roles, activities, and artifacts in the RUP Inception phase of a COTS project**

Because the Acquirer’s view is broad, the Acquirer rarely engages in detailed evaluation, recommendation, and acquisition activities, preferring to leave that to more specialized practitioners. For instance, the Acquirer may need to collaborate with a System Analyst to structure the product evaluation and with an Implementer to perform product experimentations.

**Lifecycle Objectives Milestone**

The Inception phase only ends when the objectives of the phase are reached, as described in the Lifecycle Objectives Milestone. The project may be aborted or considerably re-thought if it fails to reach this milestone. The Inception phase exit criteria are defined as follows in case of a project that involves COTS packages:

- Affected stakeholders concur that the scope of each candidate solution is feasible and represents a useful capability.
- Critical mismatches between stakeholder needs and candidate solution capabilities are negotiated and represented in critical use cases and non-functional requirements.
• Cost/schedule estimates, project tasks, risk and engineering processes for each candidate solution are credible.
• The risks for each candidate solution are understood, fall within an acceptable range, and mitigations are identified for critical risks.
• The potential changes to the end-user’s business process have been identified for each candidate solution, with stakeholder agreement to implement the changes, should the solution be selected.
• The depth and breadth of an architectural proof-of-concept (e.g., mockup, simulation) demonstrate the defined scope of each candidate solution.
• The project has initiated relationships with key vendors to provide needed insights into COTS packages capabilities and directions.
• The experimentation facility that will be used to evaluate COTS package solutions is in place and sufficient to assess the impact of the candidate solutions within the broader context of the organization’s architecture in the Elaboration phase.
• The contracting vehicles (e.g. license agreements) necessary to evaluate the COTS packages in the Elaboration phase are in place.

The Elaboration phase

The goal of the Elaboration phase is to achieve sufficient stability of the requirements and architecture; to select and acquire COTS packages; and to mitigate risks so that a single, high-fidelity solution can be identified with a predictable cost and schedule. The following sections describe the Elaboration objectives, roadmap, roles, activities, artifacts, and milestone.

Elaboration objectives

In the context of a project that aims at performing COTS package evaluation, recommendation, acquisition, installation, configuration, fielding, and evolution, the primary objectives of the Elaboration phase include:

• Insuring stability of the project's vision
• Refining the identification of the stakeholder needs and desired business processes, and insuring stability of the use cases and non-functional requirements of the solution
• Refining the characterization of the architectural and design constraints imposed on the solution by any infrastructure on which the solution will run and any other systems with which the solution must interact
• Refining the characterization of the project management constraints (time, money, people, etc.) for the project, potential risks, as well as tolerance for and inhibitors to implementing changes across the organization
• Refining the understanding of the COTS packages under consideration. Monitoring the offerings of the marketplace in terms of relevant COTS packages and vendors
• Among the candidate solutions, selecting and refining one solution that best meets the demands of the use cases and non-functional requirements, architectural and design constraints, and project management constraints with reasonable risks
• Agreeing upon and planning the changes to the end-user business processes caused by the selected solution
• Baselining the architecture of the selected solution (including COTS package integration mechanisms) and addressing all architecturally significant risks of the project
• Demonstrating that the baselined architecture of the selected solution will support the requirements of the system at a reasonable cost and in a reasonable time
• Recommending the acquisition of the COTS packages needed to form the selected solution together with the implementation of the selected solution in the Construction phase
• Acquiring the COTS packages needed to implement the recommended solution
• Refining the set up of the supporting environment for the project in the experimentation facility

Elaboration roadmap

The focus of the Elaboration phase is first on selecting, among the candidate solutions defined in the Inception phase, one feasible solution that best meets the demands of the use cases and non-functional requirements, architectural and design constraints, and project management constraints (time, money, people, scope, etc.) with reasonable risks. This is done by continuing to gather information from each of the four spheres of influence (including a detailed evaluation of the COTS packages under consideration) and by using the gathered information to refine the candidate solutions. This includes continuing to perform a gap analysis between the project needs and what is available on the market, negotiating tradeoffs with stakeholders in case of mismatches, and refining the architecture and design of each candidate solution. One goal is to capture information about the integration of the COTS packages into the acquirer environment, the necessary configuration of the COTS packages, the necessary data migration, as well as the need for custom development.

The Elaboration phase concentrates on in-depth, hands-on experiments with the candidate solutions by end-users and engineers. For each candidate solution, an executing system (called the "executable architectural prototype" in the Elaboration phase) is built in the context of the experimentation facility. The experimentation facility should replicate, as closely as possible or practical, the operational environment (including interfaces to any important external systems). A key objective of building an executing system is to better understand the complexity involved in COTS package integration, COTS package configuration, data migration, and custom development. Another goal is to continue prototyping the changes to end-user business processes in order to better understand how the COTS packages contained in each candidate solution affect the end-user business processes and make sure the stakeholders understand the impact of each solution on the business.

When the candidate solutions are sufficiently understood, one solution is selected that will become the basis for the Construction phase. The selected solution is further refined, using the experimentation facility, until it is shown that it has achieved sufficient stability in requirements and architecture and that all architecturally significant risks have been addressed. The team then makes a recommendation, based on business goals as well as system requirements and a proven architecture, to acquire the COTS packages needed to form the selected solution and to implement the solution in the Construction phase. The procurement of the COTS packages is initiated, which includes negotiating the prices and payment terms as well as obtaining the authorization of placing a purchase order with the vendor.
Figure 5 illustrates typical iterations in the Elaboration phase.

**Figure 5: Typical iterations in the RUP Elaboration phase of a COTS-based project**

Each early elaboration iteration may focus on:
- Gathering information from spheres of influence
- Refining candidate solutions
- Building an executable architectural prototype
- Recommending one or several candidate solutions to be further refined in later iterations

Iteration repeated until one candidate solution is selected that best meets:
- Business goals, use cases & non-functional requirements,
- Architecture & design constraints,
- Project management constraints with reasonable risks

Once one solution is selected, each later elaboration iteration may focus on:
- Gathering information from spheres of influence
- Refining the selected solution
- Building an executable architectural prototype
- Recommending the selected solution to be implemented in the Construction phase
- Arranging the procurement of the COTS packages forming the recommended solution

The actual number of iterations depends on the complexity and risk inherent in the desired capability. Typically, there are at least two iterations in the Elaboration phase, as illustrated in Figure 5. Early iteration(s) evaluate the candidate solutions from the Inception phase to select the best single solution, and later iteration(s) build the selected solution to a level of detail sufficient for the Construction phase to begin. Within the latter iteration(s), the selected solution is explored to mitigate risks and develop specific plans for constructing a production-quality solution. Multiple iterations may be required to mitigate the technical, programmatic, and operational risks and achieve a stable baseline of the requirements and architecture for the selected solution.

**Essential roles, activities, and artifacts in Elaboration**

This section presents some examples of roles, activities, and artifacts that are new to RUP and introduced because they are important in COTS-based projects. These elements play a significant part in the Elaboration phase (even though they may be involved throughout the project lifecycle). Refer to the Appendix for the complete list of new elements.

A new role that comes from the version of RUP for Systems Engineering is the **System Architect**. The System Architect oversees the entire system and all factors that might affect its development.
The System Architect establishes and refines the system logical and physical structure and is concerned with the optimization of these structures in terms of major system elements and their interfaces, as well as making trade-offs in competing factors and constraints.

The System Architect is responsible for performing a new activity called Define Solution. The goal is to combine the information gathered from the four spheres of influence to form a candidate solution. This is done by performing a gap analysis between the project needs and what is available on the market, negotiating tradeoffs with stakeholders in case of mismatches, and defining the architecture and design of the candidate solution.

Each candidate solution is defined within a new artifact called the Solution Dossier. A Solution Dossier aggregates, into a logical unit, all the artifacts that form the solution. For instance, a Solution Dossier may include one COTS Package Dossier per COTS package included into the solution, one Vendor Dossier per vendor of these COTS packages, as well as artifacts defining the architecture and design of the solution (like Design Model, Software Architecture Document, Package Configuration Specification, and Data Migration Specification). These are just examples of artifacts forming a Solution Dossier.

Because the System Architect view is broad and spans the entire system, the System Architect rarely engages in depth in the detailed engineering of a system, preferring to leave that to other practitioners in the various engineering specialties. Consequently, the System Architect is rarely the owner of the individual artifacts forming the Solution Dossier.

For instance, the new Package Configuration Specification artifact is owned by the Designer and produced in the context of the new Specify Package Configuration activity. The goal is to specify the COTS package configuration requirements. This involves defining the scope of the configuration effort (while having in mind the ability to upgrade COTS packages in the future), as well as identifying the COTS package parameters and their values required to define an end-user-specific version of the packages. These parameters typically relate to elements such as functionality, security and control, and reporting.

Similarly, the new Data Migration Specification artifact is owned by the Data Designer and produced in the context of the new Specify Data Migration activity. The goal is to define the scope of the data migration effort, specifying the data profile, and mapping between the data sources and the target databases, as well as identifying manual and automated data migration.

Figure 6 illustrates some of the new roles, activities, and artifacts that play an important part in the Elaboration phase.
Lifecycle Architecture Milestone

The Elaboration phase only ends when the objectives of the phase are reached, as described in the Lifecycle Architecture Milestone. The project may be aborted or considerably re-thought if it fails to reach this milestone. The Elaboration phase exit criteria are defined as follows in case of a project that involves COTS packages:

- Affected stakeholders agree that, within acceptable risk, the project’s vision can be met in the context of one solution architecture if the current plan to build the solution is executed.
- The requirements are identified and negotiated to best leverage the marketplace and other sources.
- The architecture is stable and reflects a structure that is flexible enough to support the continuing evolution of COTS packages and operational needs.
- The stakeholders, who are subject to the changes, support the necessary changes to the end-user’s business process.
- The key approaches to be used in test and evaluation are proven.
- An executing system (the “executable architectural prototype”) demonstrates the common understanding of the solution that has been achieved through negotiation with stakeholders and addresses (and resolves as appropriate) major risks of the solution.
- The remaining risks are understood, are acceptable, and have appropriate risk-management strategies identified.
- The Business Process Change Management Plan realistically accounts for all types of end-users and the necessary changes for both individuals and their organizations.
- The project has defined and implemented relationships with key vendors that provide needed insights into COTS packages capabilities and directions.
- Cost/schedule estimates and project tasks are credible.
- Any differences between actual resource expenditures versus planned expenditures for this phase are understood, and corrective actions are planned.
• The iteration plans for the Construction phase are of sufficient detail and fidelity to allow the work to proceed.
• The contracting vehicles (e.g. license agreements, contracts) for needed COTS packages and services are in place for the Construction phase and initial fielding.
• The experimentation facility is sufficient to support beginning the Construction phase.

The Construction phase

The goal of the Construction phase is to achieve a production-quality release ready for its user community. The selected solution is prepared for fielding. The following sections describe the Construction objectives, roadmap, roles, activities, artifacts, and milestone.

Construction objectives

In the context of a project that aims at performing COTS package evaluation, recommendation, acquisition, installation, configuration, fielding, and evolution, the primary objectives of the Construction phase include:

• Maintaining current information on the offerings of the marketplace as well as information on any changes to the COTS packages used in the solution
• Balancing the need for introducing changes due to the volatility of the marketplace with the need for engineering stability
• Completing the design, implementation, and testing of the selected solution
• Achieving adequate quality as rapidly as practical
• Achieving useful versions (alpha, beta, and other test releases) as rapidly as practical
• Minimizing development costs by optimizing resources and avoiding unnecessary scrap and rework
• Developing a solution that is ready to transition to its user community, which involves preparing the software, the sites, and the users for the initial fielding of the solution

Construction roadmap

The focus of the Construction phase is on preparing a production-quality release of the selected solution that is suitable for fielding. One area of work in this phase concentrates on continuing to gather information from each of the four spheres of influence to collect any information necessary to complete the Construction of the solution. Once the information is gathered from the spheres, any remaining details in the design of the solution are completed. This includes detailed design elements related to COTS package integration, COTS package configuration, data migration, any custom development, and a modified legacy component. Finally, production rigor is applied to complete the implementation and testing of the solution.

The Construction phase prepares the solution for fielding. It includes the preparation of a support strategy (that could be based on a combination of instructor-led and Web-based training, just-in-time mentoring, help desk and technical support, for instance) and design of the necessary support materials (such as training materials, installation instructions, and user manuals). Any support materials necessary to perform acceptance testing or to the initial rollout of the solution need to
be developed. The Construction phase also includes preparing the needed changes to the end-user's business processes associated with the solution. For instance, this involves restructuring the organization as necessary and developing new policies and procedures to support the solution. Any changes to the end-user's business processes necessary to perform acceptance testing or to the initial rollout of the solution are implemented.

Figure 7 illustrates a typical iteration in the Construction phase.

**Figure 7: Typical iteration in the RUP Construction phase of a COTS-based project**

There is at least one iteration in the Construction phase. The actual number of iterations required depends on the amount of work necessary to complete the design, implementation, and test of the solution, and achieve a production-quality release ready for fielding. For instance, a large amount of work related to COTS package configuration is likely to increase the number of iterations.

While every effort has been made during the Elaboration phase to stabilize the solution and to address risks, some unanticipated changes may occur in requirements, in the COTS packages themselves, and in the architecture and design during the Construction phase. In particular, because of the volatile nature of the marketplace, new versions of the selected COTS packages may require detailed investigation as vendors add, change, or remove functionality. Continued monitoring of the marketplace and evaluation of new and changed COTS packages is required to anticipate changes and determine an appropriate COTS package upgrade approach. The experimentation facility created to support the Elaboration phase continues to be needed to evaluate new and changed COTS packages. The risk to the cost and schedule of the Construction phase presented by any change has to be balanced against the risk of not upgrading and delivering obsolete COTS packages. For minor changes, the Construction phase can be temporarily delayed while adjustments are identified, validated, and implemented. For major changes, decisions made at the Lifecycle Architecture Milestone (or even at the Lifecycle Objectives Milestone) may have to be revisited or the changes deferred to the next generation of the solution.

**Essential roles, activities, and artifacts in construction**

This section presents some examples of roles, activities, and artifacts that are new to RUP and specific to COTS-based projects. These elements play a significant part in the Construction phase,
even though they may be involved throughout the project lifecycle. Refer to the Appendix for the complete list of new elements.

The Implementer (which is an existing RUP role) plays a key part in the Construction phase by being responsible for (among other things) configuring the COTS packages and migrating data. A new Implementer activity, called Perform Package Configuration, describes the process of configuring the configuration files provided by the vendor according to the Package Configuration Specification. Another new Implementer activity, called Perform Data Migration, describes the process of migrating data from the legacy databases to target databases according to the Data Migration Specification. In addition, this activity involves the validation of the data migration process and documentation of the results in the new Data Migration Evaluation artifact.

Figure 8 illustrates some of the new activities and artifacts that play an important part in the Construction phase.

**Figure 8: Roles, activities, and artifacts in the RUP Construction phase of a COTS-based project**

**Initial Operational Capability Milestone**

The Construction phase only ends when the objectives of the phase are reached, as described in the Initial Operational Capability Milestone. The project may be aborted or considerably re-thought if it fails to reach this milestone. The Construction phase exit criteria are defined as follows in case of a project that involves COTS packages:

- Affected stakeholders agree that the solution baseline, as demonstrated in the executing system, is mature enough to be fielded in the user community. The release is stable. Existing defects and pending changes are not obstacles to achieving the purpose of the release.
- Affected stakeholders agree that any restructuring of the organization necessary for the initial fielding is stable; new or modified business policies and procedures have been instituted, documented, and adopted; necessary knowledge and skills transfer has occurred; and mechanisms to encourage the solution adoption (like reward, incentive, and compensation programs) have been established.
• Relationships with vendors are adequately managed.
• Information on relevant COTS packages and the marketplace is current and recorded.
• Any differences between actual resource expenditures versus planned expenditures for this phase are understood, and corrective actions are planned.
• The iteration and deployment plans for the Transition phase are sufficiently detailed and accurate to allow the work to proceed.
• The contracting vehicles (e.g. license agreements, contracts) are in place for initial fielding and in progress for full fielding.
• The experimentation facility is sufficient to support continued monitoring of the COTS packages and relevant market segments.

The Transition phase

The goal of the Transition phase is to field the selected solution to the user community and provide necessary support. The following sections describe the Transition objectives, roadmap, roles, activities, artifacts, and milestone.

Transition objectives

In the context of a project that aims at performing COTS package evaluation, recommendation, acquisition, installation, configuration, fielding, and evolution, the primary objectives of the Transition phase include:

• Achieving final solution baselines as rapidly and cost effectively as practical. This includes tuning activities, such as bug fixing, enhancement for performance, and usability
• Achieving stakeholder concurrence that the baselines for fielding are complete and consistent with the acceptance criteria of the vision
• Achieving user satisfaction and self-supportability (e.g., procedures in place, training complete, and maintenance plan in place)
• Implementing changes to the end-user's business processes across the user community
• Maintaining current information on the offerings of the marketplace as well as information on any changes to the COTS packages used in the solution through managing relationships with vendors
• Balancing the need for introducing changes due to the volatility of the marketplace with the need for solution stability
• Positioning contracting vehicles for full fielding and long-term support of the solution

Transition roadmap

The Transition phase focuses on the integration and deployment of the solution to the user community. This requires that the end-users 1) attain proficiency in the solution and end-user business processes supported by the solution, 2) are motivated to use the solution, and 3) are self-supporting in their use of the solution.

One area of work in the Transition phase concentrates on monitoring the four spheres of influence for any changes that may affect the solution being fielded. Changes may be necessary if there are COTS package changes during the phase, or to meet the unique needs of specific installation
sites, or to accommodate change requests (such as defects and minor enhancement requests) resulting from operational experiences. In the Transition phase, user feedback should aim at fine-tuning the solution and thus focus mainly on configuration, installation, and usability issues. All the major structural issues should have been worked out much earlier in the lifecycle. Once the information is gathered from the spheres, it is used to refine the solution definition and execute the system implementing the solution.

Another area of work in the Transition phase concentrates on fielding the solution across the user base. It includes refining and implementing the support strategy as well as developing the necessary support materials (such as training and user manuals). As an example, a support strategy could be based on a combination of instructor-led and Web-based training, just-in-time mentoring, help desk and technical support, as well as mechanisms to encourage the solution adoption. Indeed, during fielding, it is common to experience resistance to implementation of the new capability. This resistance can sometimes be overcome with careful nurturing of champions for the solution among universally regarded experts, and with incentives to reward users for adopting the solution. Fielding also includes implementing the changes to the end-user business processes. For instance, some people have new responsibilities and follow new business procedures.

The Transition phase encompasses continued support for the solution. Once the solution has been fielded to the users, the focus shifts to scheduling and implementing maintenance releases of the solution. These maintenance releases are typically needed to respond to a variety of needs:

- Latent errors in the solution that may require an immediate fix
- Enhancements and more routine bug fixes that can be accommodated in more routine or periodic releases
- To incorporate COTS package patches and new COTS package releases

Planning includes determining the defects to be fixed and enhancements to be made in each release. Each maintenance release will require one or more iterations.

COTS package obsolescence and solution stability must be carefully balanced in support of the solution. The Transition phase manages the introduction of updated COTS packages while continuing to meet the demands of the operational environment. Continued monitoring of the marketplace is required to anticipate changes. Maintaining an experimentation facility for COTS package evaluation to assess the potential impact of new or changed COTS packages is essential. In some cases, new COTS package releases may require that the COTS package configuration and integration originally used to integrate the COTS packages be re-implemented.

The Transition phase ends when the solution is retired or replaced by a new generation of the solution. A new generation of the solution requires a new RUP evolution cycle (that repeats all of the RUP phases). A new RUP evolution cycle differs from a maintenance release in that the scope of the changes results in changes to the solution architecture, changes to agreed-upon end-user business processes, or a cost that exceeds the threshold for this phase.

Figure 9 illustrates typical iterations in the Transition phase.
Figure 9: Typical iterations in the RUP Transition phase of a COTS-based project

Each transition iteration may focus on:

- Gathering information from spheres of influence
- Refining the solution
- Building production-quality version of the solution
- Rolling out the solution across the user base (early iteration(s) including implementing the changes to end-user business processes)
- Providing long-term support and maintenance (later iteration(s))

The actual number of iterations depends on the complexity involved in transitioning the solution to the end-users and providing support. Typically, there are at least two iterations in the Transition phase: early iteration(s) roll out the solution across the user base; later iteration(s) focus on the long-term support of the solution. Multiple iterations may be necessary to roll out the solution across the user base, depending, for instance, on the number of end-users, their ability and willingness to embrace changes, their experience in the application domain and with the solution, as well as the number of business processes that are new or modified. Multiple iterations may also be necessary to implement the long-term support strategy of the solution. For instance, it may be necessary to plan several maintenance releases, each being composed of one or more iterations depending on the number and complexity of defects to be fixed and enhancements to be made.

Essential roles, activities, and artifacts in Transition

This section presents some examples of roles, activities, and artifacts that are new to RUP and specific to COTS-based projects. These elements play a significant part in the Transition phase (even though they may be involved throughout the project lifecycle). Refer to the Appendix for the complete list of new elements.

A new artifact (that has been created in the Inception phase and updated in subsequent phases) that plays a key role in the Transition phase is the Business Process Change Management Plan, which is owned by the Project Manager and created in the context of a new activity called Develop Business Process Change Management Plan, as illustrated in Figure 10.
The goal is to create an effective and documented plan to increase the likelihood of successful implementation of end-user business process changes and to provide a structured approach to managing the human elements critical to achieving strategic business objectives. One key goal of these new activities and artifacts is to ensure that business process change management is recognized -- at all organizational levels -- as a process that touches many of the socio-technical activities at work in the organization and not just as an isolated activity. However, in order to fully address the management of very complex business re-engineering efforts (under which circumstances the COTS package project may be a secondary consideration), the Business Process Change Management Plan may need to be subsumed in a set of dedicated artifacts and related business re-engineering processes not covered in this RUP configuration. Our intent here is to address business process change only to the extent that it is necessary for successful COTS package solution deployment.

**Product Release Milestone**

The Transition phase ends when the objectives of the phase are reached, as described in the Product Release Milestone. The Transition phase exit criteria are defined as follows in case of a project that involves COTS packages:

- Solution functionality is no longer needed or is replaced by a new generation of the solution.
- Lessons learned as well as improvements to and other useful information on the notations, processes, methods, and tools used during the COTS-project are collected and made accessible to future solutions or projects.
- Information is reviewed, analyzed, and used to improve the organization's standard notations, processes, methods, and tools.

**Conclusion**

The RUP for COTS implements a variety of best practices for COTS-based projects, which take into account the unique characteristics of COTS packages. For instance, it proposes an iterative approach where COTS packages (and therefore, the marketplace) drive the evolving definition of the solution. This approach helps IT teams develop and maintain a resilient system architecture as a centrally managed asset, which balances COTS package obsolescence and solution stability, and which incorporates activities for changing the end-user business processes.
Refer to the RUP for COTS Website for a complete presentation of the process. For each RUP phase, this Website provides a detailed description of the objectives, milestone, roadmap, roles, activities, and artifacts, as well as sample iteration plans.

To download a PowerPoint presentation describing RUP for COTS, click here. In the "Download" box that appears, you will see the filename "rup4cots_intro.zip"; to begin the download, click on the word "FTP" at right.

Acknowledgments

The authors would like to acknowledge the contributions of all the reviewers, and thank them for their insightful comments on early drafts of this paper. Especially, they would like to warmly thank Lisa Brownsword from CMU/SEI. Her enthusiasm for the creation of a RUP version of EPIC, her support of the project, as well as the time she spent sharing her experience in EPIC, contributed greatly to the success of the project.

Further reading:


Appendix: New elements in RUP for COTS

This appendix summarizes, for each RUP discipline, the elements of RUP for COTS that are specific to or important in COTS-based projects. Refer to the RUP for COTS Website for a detailed description of each of these elements.
<table>
<thead>
<tr>
<th>Discipline / Workflow Detail / Activity</th>
<th>Role</th>
<th>Output Artifact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discipline: Project Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow Detail: Plan the Project</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow Detail: Acquire Package*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Arrange for Procurement of Package*</td>
<td>Acquirer*</td>
<td>Contracting Vehicle*</td>
</tr>
<tr>
<td>Activity: Recommend Solution*</td>
<td>Acquirer*</td>
<td>Solution Recommendation*</td>
</tr>
<tr>
<td>Activity: Manage Receipt of Package*</td>
<td>Acquirer*</td>
<td>Package Acceptance Document*, Vendor Deployment Plan*, Vendor Deployment Unit*</td>
</tr>
<tr>
<td>Workflow Detail: Monitor and Control Vendor*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Manage Request for Information*</td>
<td>Acquirer*</td>
<td>Vendor Response*, Request for Information*</td>
</tr>
<tr>
<td>Activity: Manage Request for Proposal*</td>
<td>Acquirer*</td>
<td>Vendor Response*, Request for Proposal*</td>
</tr>
<tr>
<td>Activity: Manage Contracting Vehicles*</td>
<td>Project Manager</td>
<td>Contracting Vehicle*</td>
</tr>
<tr>
<td>Activity: Manage Vendor Relationships*</td>
<td>Acquirer*</td>
<td>Vendor Management Plan*, Vendor Dossier*</td>
</tr>
<tr>
<td>Discipline: Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow Detail: Analyze the Problem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Identify Relevant COTS Package and Vendors*</td>
<td>Acquirer*</td>
<td>COTS Package Dossier*, COTS Package Screening Criteria and Rationale*, Market Segment Information*, Vendor Dossier*</td>
</tr>
<tr>
<td>Workflow Detail: Understand Stakeholder Needs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Identify Relevant COTS Package and Vendors*</td>
<td>Acquirer*</td>
<td>COTS Package Dossier*, COTS Package Screening Criteria and Rationale*, Market Segment Information*, Vendor Dossier*</td>
</tr>
<tr>
<td>Workflow Detail: Define the System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Identify Relevant COTS Package and Vendors*</td>
<td>Acquirer*</td>
<td>COTS Package Dossier*, COTS Package Screening Criteria and Rationale*, Market Segment Information*, Vendor Dossier*</td>
</tr>
<tr>
<td>Activity: Produce Request for Information*</td>
<td>Acquirer*</td>
<td>Request for Information*</td>
</tr>
<tr>
<td>Activity: Prepare for Vendor/Package Evaluation Workshop*</td>
<td>Acquirer*</td>
<td>Evaluation Score Sheet*, Vendor and Package Evaluation Workshop Description*</td>
</tr>
<tr>
<td>Activity: Conduct Vendor/Package Evaluation Workshop*</td>
<td>Acquirer*</td>
<td>Evaluation Score Sheet*</td>
</tr>
<tr>
<td>Workflow Detail: Refine the System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Characterize Relevant COTS Package and Vendors*</td>
<td>Acquirer*</td>
<td>COTS Package Dossier*, COTS Package Screening Criteria and Rationale*, Market Segment Information*, Vendor Dossier*</td>
</tr>
<tr>
<td>Activity: Produce Request for Proposal*</td>
<td>Acquirer*</td>
<td>Request for Proposal*</td>
</tr>
<tr>
<td>Activity: Prepare for Vendor/Package Evaluation Workshop*</td>
<td>Acquirer*</td>
<td>Evaluation Score Sheet**, Vendor and Package Evaluation Workshop Description*</td>
</tr>
<tr>
<td>Activity: Conduct Vendor/Package Evaluation Workshop*</td>
<td>Acquirer*</td>
<td>Evaluation Score Sheet*</td>
</tr>
<tr>
<td>Workflow Detail: Manage Changing Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity: Monitor COTS Packages and Vendors*</td>
<td>Acquirer*</td>
<td>COTS Package Dossier*, Market Segment Information*, Vendor Dossier*</td>
</tr>
</tbody>
</table>

**Discipline: Analysis and Design**

<table>
<thead>
<tr>
<th>Workflow Detail: Perform Architectural Synthesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity: Define Solution*</td>
</tr>
</tbody>
</table>

**Workflow Detail: Define a Candidate Architecture**

| Activity: Define Solution*                        | System Architect* | Solution Dossier* |

**Workflow Detail: Refine the Architecture**

| Activity: Define Solution*                        | System Architect* | Solution Dossier* |

**Workflow Detail: Design Components**

| Activity: Specify Package Configuration*          | Designer          | Package Configuration Specification* |

**Workflow Detail: Design the Database**

| Activity: Specify Data Migration*                 | Database Designer  | Data Migration Specification* |

**Discipline: Implementation**

<table>
<thead>
<tr>
<th>Workflow Detail: Implement Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity: Perform Data Migration*</td>
</tr>
<tr>
<td>Activity: Perform Package Configuration*</td>
</tr>
</tbody>
</table>

**Workflow Detail: Integrate Each Subsystem**

| Activity: Perform Data Migration*                | Implementer       | Implementation Elements |
| Activity: Perform Package Configuration*         | Implementer       | Implementation Elements |

**Workflow Detail: Integrate the System**

| Activity: Perform Data Migration*                | Implementer       | Implementation Elements |
| Activity: Perform Package Configuration*         | Implementer       | Implementation Elements |

**Discipline: Environment**

<table>
<thead>
<tr>
<th>Workflow Detail: Support Environment During an Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity: Install System*</td>
</tr>
<tr>
<td>Activity: Set Up and Maintain Experimentation Facility*</td>
</tr>
</tbody>
</table>

**Notes**


Ibid.


6 Albert and Brownsword, Op cit.


8 For more information on progressive acquisition of COTS packages, see Max Wideman's five-part series, "Progressive Acquisition and the RUP" published in The Rational Edge between December 2002 and April 2003:
   Part I: Defining the Problem and Common Terminology
   Part II: Contracts That Work
   Part III: Contracting Basics
   Part IV: Choosing a Form and Type of Contract
   Part V: Contracting Activities

© Copyright IBM Corporation 2005
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)