Reuse and Return on your Analysis Investment

First proposed by Ivar Jacobson, the use case has become commonplace in today's technology projects. Their widespread use can be attributed to the fact that use cases are effective tools in the early stages of system development (the analysis phase) and help stakeholders gain a greater understanding of a system's functionality as well as its interaction with actors and other systems. This functionality alone has made use cases a preferred tool among business analysts to define scope and move a project forward into development.

Yet even with their popularity uses cases are widely underutilized, rarely being developed beyond the initial analysis phase. Given the level of resources and the importance of the information that is captured in use cases, to allow them to just collect dust can be a costly misstep. Use cases can add value well past the analysis phase by contributing to tasks such as testing, maintenance and continuous improvement, and planning for future development. But to gain continuous returns from the use case investment, analysts must understand that it is possible to utilize use cases in many ways. A multi-level approach to use case development makes it possible to achieve additional benefits and added functionality. Each level is specifically designed to capture pertinent information that reflects the needs of a project at its different phases within the system development life cycle.

Continuous Added Value

If maintained and expanded as the development of the system progresses, use cases can provide value at all stages of a project's lifecycle. The use case must be broken down into three separate stages; each stage reflects a particular phase in a project's lifecycle where the use case can be leveraged to capture pertinent information. Breaking use cases down in this manner best serves the needs of a project at all phases of development and facilitates the transition from phase to phase. Equally important, the use cases should be documented not just in text, but in a modeling tool that offers an underlying repository. With the support that such a tool offers can the analyst easily capture and expand the necessary information.

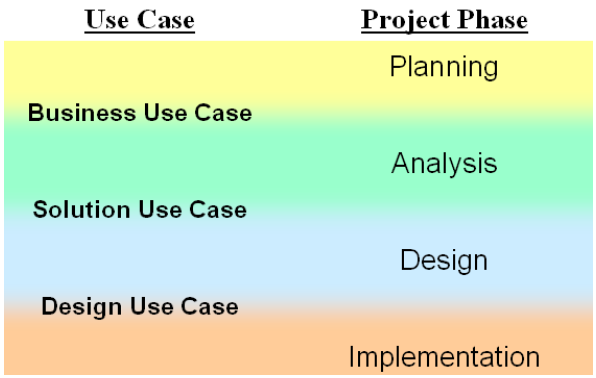

Figure 1—Project Phase Mapping

The three stages of use case development are: the Business Use Case, the Solution Use Case and the Design Use Case. Each stage of the use case is tailored to accomplish a particular goal using a recommended set of products which build upon those of the previous stages, while acting as a gateway to the next.

The Business Use Case – a Project's Starting Point

Laying out the objectives and desired results at a project's inception is critical to its success. Business Use Cases are developed by business analysts in conjunction with project sponsors and then used as communication tools by all stakeholders in order to gain a clear understanding of the goals of the project and the functionality of the system.

This information is captured clearly in a set of preliminary use cases descriptions and diagrams which lay the groundwork for the project and are decomposed later in the project by including additional detail as it is discovered using business analysis and elicitation techniques.

To develop the Business Use Case the project team will need to take steps to: analyze and document business needs, the characteristics of the organization, document

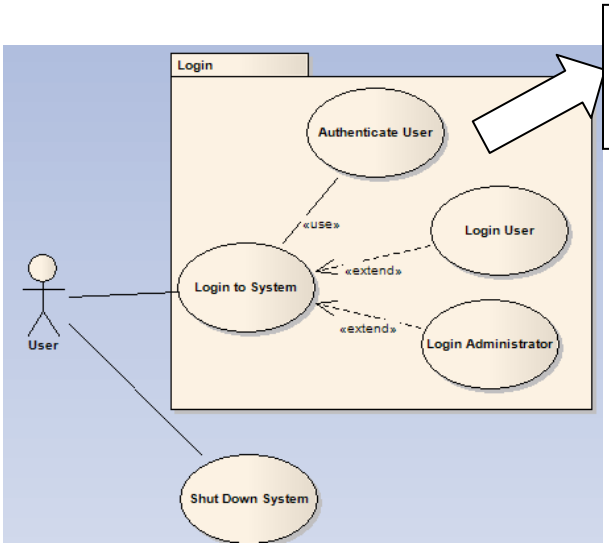

Figure 2— Business Use Case

the as-is state, develop a preliminary design of the ideal to-be state, and consider what impact these changes will have. To accomplish theses tasks, techniques such as business process modeling, business event review, and actor/user identification are useful tools to flush out these critical aspects of the project. The final output for the business use case is a set of identified use cases and basic descriptions for each use case which will be detailed in later stages. To diagram the set of identified use cases, descriptions, actors and relationships, the Unified Modeling Language (UML) Use Case Diagram is recommended. Figure 2 provides an example Use Case Diagram which depicts the basic functions of logging a user into a system and would have a description of each use case stored in the modeling tool. Note that the use cases are grouped in a "Login" package. Upon completion, the Business Use Cases must be published for review by all stakeholders before it is used as a starting point for further development.

Solution Use Case – Discovering Details through Analysis

Once objectives, functions and desired outcomes are documented a more detailed solution must begin to take shape. Starting from the business use case and the associated artifacts, the Solution Use Case further decomposes the to-be solution adding detailed information on the system, processes, technology, and stakeholder interaction. The end goal of the Solution Use Case is to have a total view of the system steps that can be used as a tool to identify functional and non-functional requirements and move towards development. This complete view of the solution is achieved by: documenting the desired scope, specifying actor and system tasks along with relationships, and developing an implementation plan.

To capture the detailed interaction between the system and

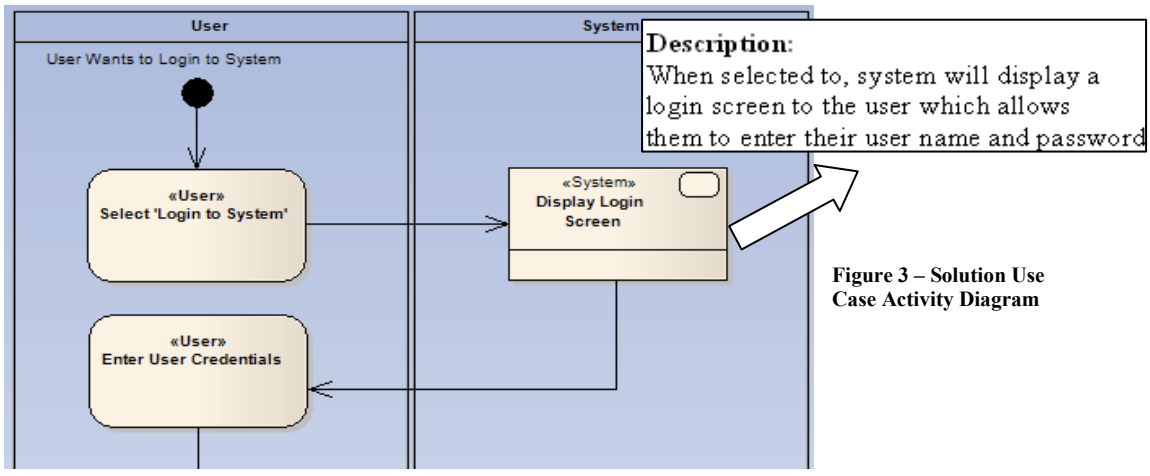

Figure 3 – Solution Use Case Activity Diagram

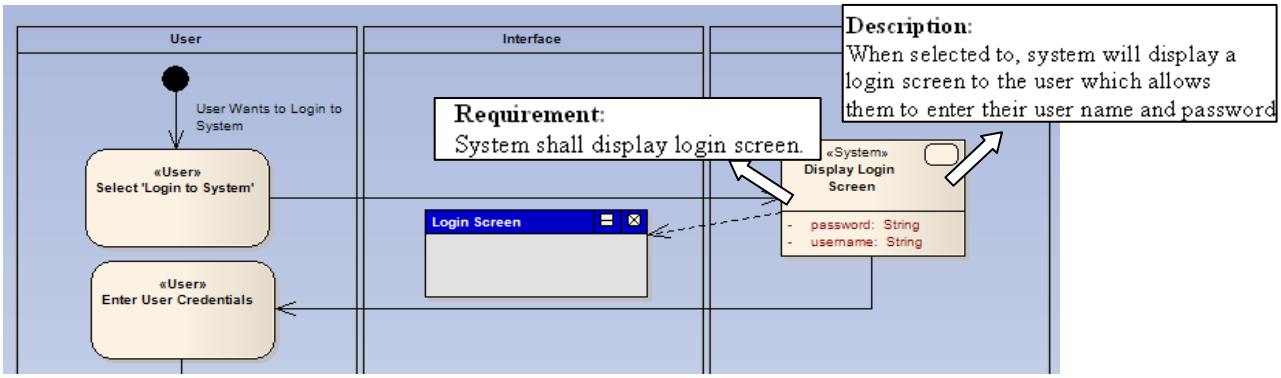

Figure 4—Design Use Case

actors the UML Activity Diagram (Figure 3), in conjunction with a requirements repository, is a useful tool to document the interworking of the proposed solution. In order to properly document and understand the proposed solution projects should look to a single tool which is capable of capturing multiple artifacts in a single repository. This one tool approach not only allows for cost savings, but also simplifies analysis and documentation efforts.

By using the UML Activity diagramming convention, analysts are able to include details such as pre/post-conditions, alternate and exceptions flows, as well as initial/final states. In the example provided the login use case begins to be decomposed into steps documenting the desired solution as an activity diagram. We use Activity symbols for the steps the user will perform and Action symbols for the steps the system will perform. This choice allowed us to link pre-defined requirements to the system steps, a feature available in the Enterprise Architect modeling tool (see the Notes area below).

Once the material is complete, a second review of the Solution Use Cases must be held which includes all relevant stakeholders and serves as a detailed system functionality review. At this point in the lifecycle the design has achieved greater stability and we recommend that a management strategy for the use cases and their corresponding requirements be considered in order to manage versioning and change requests as the project progresses and development begins.

Design Use Case – Depicting True System Design

A paradigm shift from the traditional use case, the Design Use Case documents the true design of the system as it has taken shape during detailed requirement elicitation, data analysis, UI design, interface analysis and development. Leveraging the work done in the Solution Use Case, the Design Use Case serves to further decompose and finalizes functional and non-functional requirements of the system, includes a data design for the to-be which is associated to the system activities where that data is used, and can contain detailed development information such as user interface design.

To capture information required by the design use case, the previously developed activity diagrams are leveraged and expanded upon. As shown in Figure 4, the activity diagram contains added system design information including a screenshot, detailed requirements, and data elements. While not standard UML, these added features allow for a more complete documentation of the system as it has been delivered. Again, we used the Enterprise Architect modeling tool because it allowed us the ability to easily add the features we needed.

The value of the design use case is that it captures the true final design of the system. Once complete, this well documented final design is an indispensible resource which can

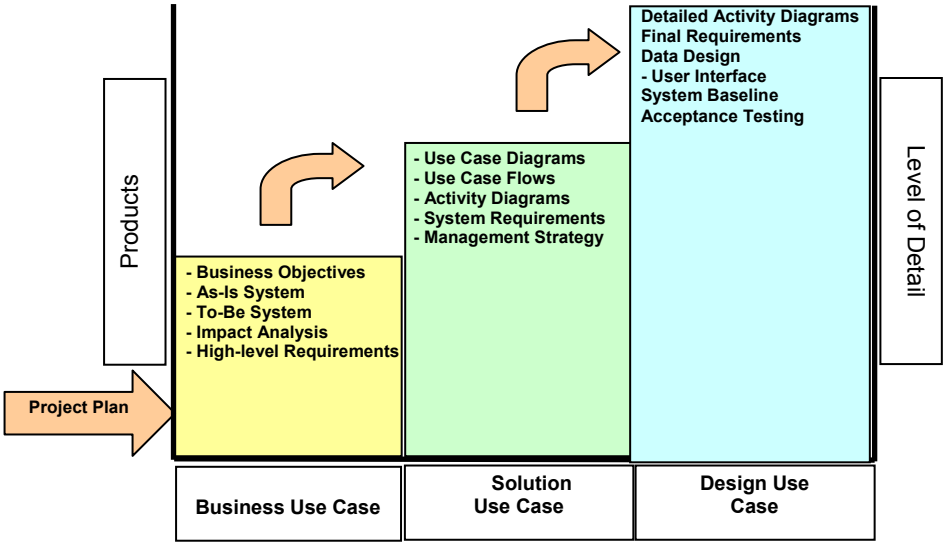

Figure 5—Use Case Progression

be used during testing for the development of artifacts such as User Acceptance Test scripts. Additionally, an overall view of the developed system can serve as a baseline in order to plan and analyze future development efforts.

Summary

Moving beyond the traditional use case model and towards a multi-level approach allows for projects to continuously gain added value from use cases at all stages of the system development lifecycle. By capturing the right details, at the right time, and organizing those details in a way that is familiar and easy to understand it is possible to improve project communication, capture design details in a meaningful way and continuously reap value from work products. This added value is particularly apparent during the latter stages when design use cases can be leveraged for acceptance testing, through the use of automatic scenario generation, and future system development. The life of the use case does not have to end with analysis. With proper structuring, good development practice, and an eye towards reuse, use cases can live on as critical tools for project success. ■

Notes: The figures in this paper were created using *Enterprise Architect*, a UML-based modeling tool from Sparx Systems, enhanced by Doreen Evans Associates to support the many roles the business analyst plays throughout a project and across an organization. Doreen Evans Associates has chosen to use Enterprise Architect because it is flexible, robust and affordable. The tool's adaptability and flexibility, along with its full repository, make it an ideal tool for our business analysts to use.

Doreen Evans Associates (DEA), a woman-owned professional services firm in business since 1992, provides a comprehensive, architecturally framed approach to the requirements challenge. DEA has developed a Center of Requirements Excellence which provides the infrastructure necessary to support requirements best practices. The infrastructure includes LINKProcess™, a detailed, defined requirements life cycle process; a suite of integrated tools to support the process; a set of templates, guidelines and macros to generate quality requirements deliverables; and a BA competency education program. Using this infrastructure and its seasoned consultants, DEA can help clients change a business process, build an enterprise architecture, or define and manage requirements for systems.