Dell re-architecture methodology

Dell’s re-architecture is an approach to application re-development that leverages existing business logic, data and application usage to fully exploit modern technologies. An engagement makes use of Dell’s re-architecture methodology and software. The methodology is a three-phase approach that includes three different tracks of activities (Figure 1).

Phase 1: Extraction—present case assets analysis

Phase 1 consists of building an inventory of existing business functionality as well as system and data components (e.g. capabilities, use cases, screens, reports, interfaces, batch processes, programs, data entities) known as the present case assets or PCA.

Business process modernization begins with business analysts documenting the user interactions with the system from the recorded walkthroughs of usage scenarios provided by actual system users. The PCA use cases and PCA business processes, also known as ‘semantic’ assets, typically include extended and re-purposed uses of the applications (e.g. spreadsheets or re-defined data elements) that are workarounds to limitations in the legacy application. The assets captured at this business process layer are linked to assets of the application layer. This allows Repository users quick access to the logic embedded in the programs and data supporting the functionality described.

Application modernization is performed by systems analysts who inventory and parse the program source code and table/file (entity) definitions of the application into ‘mechanical’ assets. The result is program-to-program call traces down to the entity-attribute layer. Analysis scripts, software features within the Dell Analyzer, are run against the parsed code to further refine call traces, identify duplicate code, etc.

Figure 1: Dell re-architecture methodology overview
highlight language specific syntax, and identify areas of potential business logic. Legacy batch functionality and system interfaces are also analyzed and documented with the associated program relationships.

**Data modernization** is performed by data modernization specialists, who inventory and profile the data entities and properties to create a logical domain model (LDM). This model expresses the legacy file or data structures in familiar object-relational terms i.e. as entities, relationships between entities, cardinalities and constraints. The LDM becomes the starting point of the transformation to a modernized domain model in the next phase.

All documentation is created within the Dell Repository (software that is part of the Dell Re-architect suite) and at the end of Phase 1, the Repository can be used to interactively follow a business process down to a use case step, through various program calls, to the entities that support the functionality being examined.

**Modernization planning** conducted before the conclusion of Phase 1, consists of a series of targeted requirements-gathering interviews to establish streamlining patterns, architecture patterns, usability requirements, non-functional requirements, and enhancement scope. While Dell Re-architect limits risk by limiting enhancements, modernization planning captures and prioritizes potential enhancements to identify low-risk opportunities that can be incorporated into the project as ‘streamlining’ - modern architectures can inherently supports certain types of enhancements - and anticipates design decisions that could simplify enhancements planned for the future.

**Phase 2: Transformation—future case assets analysis**

Phase 2 is focused on defining target models for the modernized application (screen models, service models and domain models). The ability to generate code from the screen and domain models accelerates the design and implementation process. Each model is also traceable to the equivalent PCA asset making its underlying legacy logic available as additional detailed specification in Phase 3.

**Business process modernization** in Phase 2 refers to the transformation of the legacy functional assets to their modernized equivalents (e.g. PCA use case to FCA use cases). Modern user interface patterns are also applied to the new screen designs. Streamlining opportunities identified during modernization planning are incorporated into the new design and specified in the transformed future case assets (FCA) i.e. FCA business processes, FCA use cases, or FCA screens.

**Application modernization.** An application architecture document (AAD) is created that is consistent with the customer’s enterprise architecture and addresses all elements of the application stack. The document also addresses non-functional requirements in the infrastructure design such as performance, reliability, and security.
The batches and system interfaces are transformed into FCA system events using patterns described in the AAD. Common services are defined within the FCA services asset with mapping back to legacy logic where applicable. As with business process modernization, streamlining opportunities from modernization planning are also incorporated into the FCA system events and FCA services design.

Data Modernization. The PCA logical domain model (LDM) is transformed to a FCA LDM by using the Dell Data Workbench and best-practices guiding the transformation from legacy tables (or files) into an object-relational data layer. This process also creates mapping rules that will be used later in the conversion of legacy data into modernized data within the modernized data structures.

Phase 3: Implementation—development

The key objective of this final phase is to implement the FCA design into a re-architected application. Implementation includes code generation, collaborative development, testing and data migration.

Code generation. Development iterations rely on the Dell Code Generator software, which enables the generation of the plumbing code as well as automated unit tests from the models created in phase 2. Code generation templates can be customized to fit specific application architecture requirements, improving developer productivity and optimizing best-practice coding patterns.

Collaborative development. The implementation team produces the remaining code by utilizing the Dell Analyzer to inspect the legacy source code and using analysis scripts and annotations to identify key business logic for inclusion in the framework provided by the generated code.

The Dell tools leverage continuous integration to accelerate development work and enable collaboration across a large team and multiple locations.

Testing and data migration. The methodology includes a comprehensive set of testing procedures to ensure both functional completeness of the modernized application and quality assurance of the migrated data. Dell Data Workbench enables iterative cycles of differential system testing (a.k.a. parallel testing) to ensure equivalency between the legacy and the modernized results.

Dell Re-architect suite

Dell Re-architect suite consists of a suite of products that assist and automate critical aspects of a re-architecture project.

- Dell Repository. This web-based Repository enables the project team to document and understand all the assets in the legacy application. This includes both the legacy PCA as well as transformed FCA design assets. Mechanically loaded and parsed assets like program source and data/file definitions are linked to semantic assets like use cases and business processes.
• **Dell Analyzer.** The Analyzer provides an interactive way to trace the results of the parsing and linking. For example, users can follow one business process, into a use case, into a screen, into the program that enable the screens, program-to-program call traces, and finally into program-to-entity calls. Code-level features of the Analyzer allow users to write scripts to automatically identify code blocks for potential business logic, to automatically create annotations that represent business logic, or to manually create annotations around any block of code.

• **Dell Designer.** The Designer allows users to graphically model the FCA logical domain (entities and their relationships), and FCA logical interface models (e.g. screens, reports and system interfaces). FCA services can also be modelled with specific references back to legacy code annotations for operation logic.

• **Dell Code Generator.** Code can be generated from the FCA models into a services-based architecture to offer a productivity boost to the development team and help uniformity in the target source code to allow future maintainability.

• **Dell Data Workbench.** The Data Workbench supports the migration of data throughout the project lifecycle, until the final production cutover. It uses the PCA and FCA logical domain models, and the related mapping file to migrate data from the legacy to the production environment.

---

**The Dell Repository** provides an environment for managing requirements and assets allowing Dell to reduce risk and accelerate timelines to deliver a modernized system.

**The Dell Analyzer** reverse engineers requirements by capturing how users actually use the software and discovering the connections between application functionality, code and data in an interactive and collaborative environment.

**The Dell Designer** creates modern application models based on legacy requirements discovered with Dell Analyzer, and incorporate necessary enhancements.

**The Dell Code Generator** generates code from Dell Designer into a service-based architecture, boosting productivity and quality.

**The Dell Data Workbench** allows for the design and frequent migration of data throughout the project.

The Dell Re-architect Solution doesn’t just modernize the code, it also migrates the data—assuring that all of the features your organization depend on remain intact and complete.