Access Data
CONSULTING CORPORATION

Lifecycle Methodology
White paper

Version: 1.2
Dated: February 9, 1999
Author: Software Engineering Process Group (**SEPG**)

Copyright © 1999 Access Data Consulting Corporation

What is a Software Development Methodology

A software development methodology defines a systematic approach to software development. It describes the procedures and framework team members utilize to produce business software. Although there are many software development methodologies – some are more effective than others. In the final analysis, there is only one way to measure the effectiveness of a software development methodology — that is to empirically observe how well the software produced by that methodology meets the needs of the targeted business.

The Access Data Lifecycle Methodology utilizes a "phased iterative" approach to software development. This approach incorporates the speed of Rapid Application Development (RAD) with the software quality assurance inherent in spiral development.

The Access Data Lifecycle methodology has been crafted and refined by Access Data personnel for over a decade. Our methodology extends past the traditional development boundaries to encompass system's development on an enterprise scale. The focus of the methodology is to ensure that all software development maps directly to business needs. Because of this focus, systems developed by Access Data, utilizing the Access Data Lifecycle, have become the cornerstone of many highly visible and successful business ventures.

Why does Access Data use a Software Development Methodology

Access Data is committed to producing software that meets the highest of standards. The software system must fully meet client business requirements; be delivered on time; have no defects; and include full documentation. When one examines projects that utilize a carefully observed methodology and those that do not, there is a notable difference in compliance to these key metrics of project success. By utilizing the Access Data Lifecycle, our development teams consistently outperform the remainder of the industry.

Access Data Lifecycle Methodology Description

The Access Data Lifecycle methodology utilizes a "phased iterative approach" to software development. The software development effort is broken into a series of phases (planning, analysis, design, construction, systems test, and implementation). The figure below graphically illustrates these phases. Each phase defines the project focus during that particular stage of the development effort. For instance, during the analysis phase, team members focus on business process analysis, requirements definition, functional interface specification, etc. This does not mean that analysis is the only project activity underway, just that the focus of the project's activity is analysis. When the project enters subsequent phases, analysis continues as a secondary or tertiary effort. A description of the activities performed during each project phase follows.

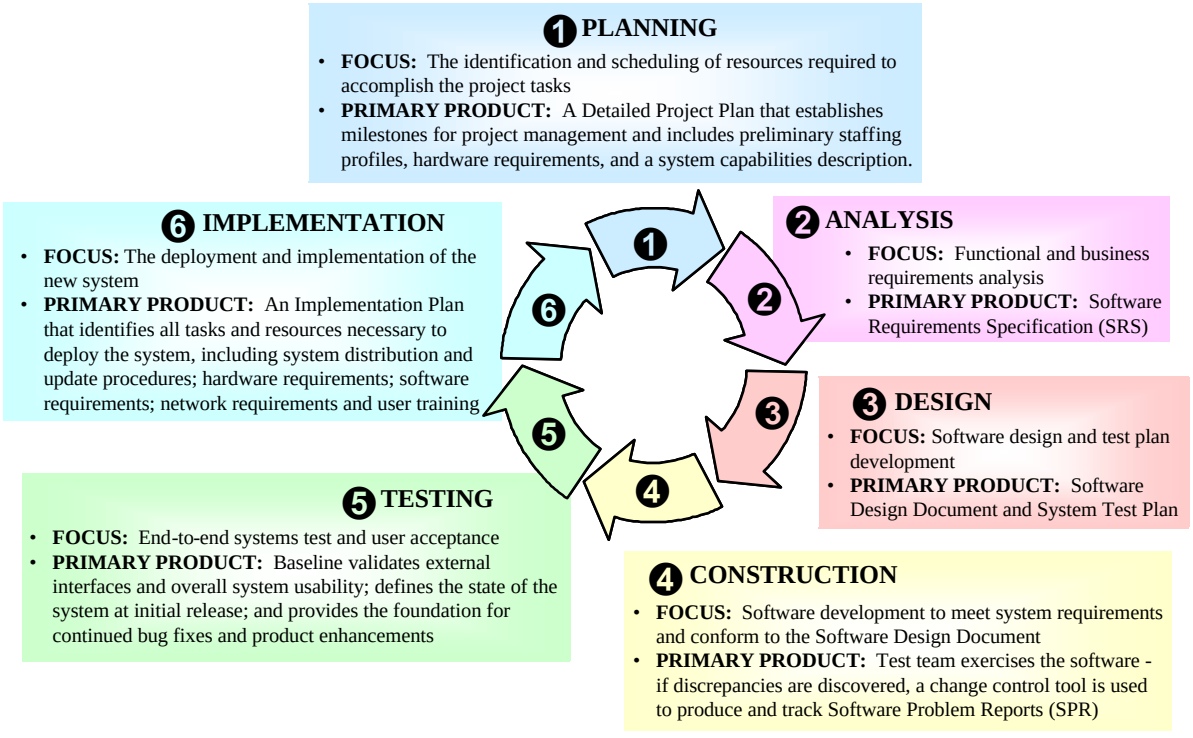
Advantages of the Access Data Lifecycle Methodology	
✓	Phased approach allows for pre-defined formal Quality checkpoints.
✓	Early involvement by the test team insures that user requirements are met.
✓	Iterative development process combines the best of both spiral and Rapid Application Development (RAD) methodologies to allow for faster delivery of a high-quality product.
✓	12 week delivery cycles provide for quick "Time To Market."

Planning Phase

The project focus during the planning segment is to identify and schedule the resources required to accomplish the project tasks. A detailed project plan is developed along with preliminary staffing profiles, hardware requirements, and system capabilities description. Often times a system functional description and/or a systems operational concept document are developed. The planning segment of the project reemphasizes the tasks and task steps established by the Access Data Lifecycle methodology.

Specifically, the planning segment defines the work breakdown structure, determines documentation requirements, describes standards for project development, and establishes milestones for project management and project control. The project control segment of the Access Data Lifecycle methodology is designed to help project managers define the procedures to start a project, track, report, and review project progress. It involves the continuous assessment of planned versus actual performance.

At the conclusion of the planning phase, the project plan is placed under change control.



The Access Data Lifecycle Methodology

Analysis Phase

The project focus during the analysis phase is functional and business requirements analysis. Emphasis is placed on generating project documentation describing the project requirements and functionality. System definitions and interface requirements to surrounding subsystems are thoroughly documented and are placed under configuration change control. These documents will later be used to establish the appropriate baseline for software configuration management. At this point in the project, the test team is assembled and requirements based test planning begins.

The primary product of the Analysis Phase is a Software Requirements Specification (SRS). At the conclusion of the analysis phase, a joint (user/customer/developer) software requirements review is conducted. This review allows the project team an opportunity for a final review and correction of the project requirements. After the final set of project requirements is agreed upon, the SRS is placed under change control.

Design Phase

The project focus during the design phase is software design and test plan development. The system requirements are decomposed into software components. The lead software engineers are brought on board; software objects are designed; the system database is described; the system process model is built; external and internal interface specifications are written; and software components are allocated to executable images. In conjunction with the software design effort, the test team develops test plans and procedures; identifies testable external interfaces; defines software stubs and drivers; and develops requirements-based test plans.

The design phase culminates in a Software Design Review followed by a System Test Review. The purpose of the Software Design Review is to solicit feedback from project personnel concerning the design of the system. The purpose of the Software Test Review is to solicit input from project personnel concerning the testing of the system. The results of the design phase (and subsequent reviews) are published in the Software Design Document and System Test Plan.

At this point in the project, the System Test Plan is put under change control. Controlling modifications to the test plan ensures that the measurement of system functionality does not change without project team acceptance.

Construction Phase

The project focus during the construction phase is software development. The software developers work with the business analysts and testers to develop software that meets system requirements and conforms to the Software Design Document. Software is developed in an iterative manner with the developers turning over new modules and bug fixes on a weekly basis. After each delivery, the test team exercises the software to ensure that it meets system requirements. In the event that discrepancies are discovered, a change control tool is used to produce and track Software Problem Reports (SPR). The appropriate developer is assigned the Software Problem Report (SPR), thereby allowing the efforts to fix the software defect to be coordinated and tracked. It is the development team's responsibility to correct software discrepancies and deliver fixes in conjunction with the new functionality scheduled for the current week's build. The construction phase is a proven and repeatable process that is an integral component to the Access Data Lifecycle methodology.

By following this iterative development methodology, the system functionality is delivered and tested a piece at a time. This allows us to identify and prevent failures early on that can occur in the integration phase of a traditional software development methodology.

At the conclusion of the construction phase, the project code, scripts, database definitions and operational environment is frozen. Tighter change control procedures are put in place, ensuring that no system changes can take place without project management concurrence. The purpose of this increased control is to provide a stable baseline from which to begin a system test.

System Test Phase

The project focus during the system test phase is end-to-end systems test and user acceptance. Throughout the previous phase, the test team has been focused on requirements based testing. During this phase, the focus is on validation of external interfaces and overall system usability. The software is run as an integrated set of functionality and requirements are re-verified in a complete system context.

During systems test, software and system changes are tracked as closely as possible. Regression testing after each change is given increased attention. The end result is to minimize the addition of new bugs into the system while allowing legitimate software fixes to proceed.

At the conclusion of the system test, a new software baseline is established via a code freeze. This baseline defines the state of the system at initial release. In addition, this baseline provides the foundation for continued bug fixes and product enhancements.

Implementation Phase

The project focus during the system implementation phase is deployment and implementation of the new system. An implementation plan is developed which describes all of the tasks and resources necessary to deploy the system. The plan normally includes system distribution and update procedures, hardware requirements, software requirements, network requirements and user training.

The Bottom Line

The Access Data Lifecycle includes all of the key practices described by the Software Engineering Institute to define a mature, well-managed software process. Through a focused and sustained

effort, Access Data has built a process infrastructure of software engineering and management practices that have proven to be highly effective for business system development. Simply put, the Access Data Lifecycle rapidly produces an on-time deployment of fully functional business systems.