

: What method is best for building an enterprise-wide, data-driven DSS?

by Dan Power

Editor, DSSResources.com

Agile, SDLC, rapid prototyping, Spiral, RAD? Or some combination of methods? Information Technology managers have strong opinions about development methods.

Using a methodology like the Systems Development Life Cycle (SDLC) provides one way business organizations can systematically approach the development of a decision support database and decision support end user capabilities. Once requirements are fixed, SDLC limits changes and reduces development flexibility. What method is best?

Development of a large, shared enterprise-wide, data-driven decision support system is usually an undertaking of great complexity.

Organizational decision making information needs are varied and needs often change in response to problems and new tasks. Some flexibility in the development process is usually needed.

Gathering and organizing corporate historical data and relevant external data so people can access the data and derived information creates both technical and organizational problems. The method used should help developers resolve both types of problems.

In general, building an enterprise-wide DSS that queries a data warehouse should proceed incrementally, but developers should follow a structured, planned approach. An incremental approach allows builders the opportunity to get feedback on prototypes of capabilities while reducing the risks associated with building an enterprise-wide system.

Inmon (2002; 2003) suggests creating the overall data warehouse conceptual plan and then implementing the subjects that are most important to stakeholders. His spiral development method is based upon incremental commitment.

The following steps for building an enterprise-wide, data-driven DSS are based upon prescriptions of consultants and case studies.

: What method is best for building an enterprise-wide, data-driven DSS?

Step 1: Decision-oriented diagnosis and definition of requirements. What are the current decision processes that most need data-driven decision support and what are the information needs?

Step 2: Feasibility analysis of a data warehouse and business intelligence decision support system. Do we need and can we afford a system? Will we benefit?

Step 3: Investigate current data. Examine the enterprise data model if it exists. Identify and describe the structure of data in current operational systems.

Step 4: Create a plan for how historical data will be organized. This is the process of identifying, modeling and documenting the data requirements of the system being designed. The data are separated into topics about which managers need information. Then developers define how the topics are related.

Step 5: Create tables and relations, and move data into the tables from operational or external systems.

Step 6: Create demonstration decision support applications for a few targeted users.

Step 7: Have targeted users test the capabilities and provide feedback.

Step 8: If justified, incrementally expand the DSS project and return to Step 5.

This is a flexible, "agile" method without stark distinctions from other structured methods like the Systems Development Life Cycle (SDLC) or Waterfall approach. The above 8 step approach should:

: What method is best for building an enterprise-wide, data-driven DSS?

1. Encourage collaboration and interactions among the development team and potential users and encourage use of processes and tools.
2. Insure that the DSS applicatoions work as intended, but promote adequate documentation. Both internal and external documentation insure that users and maintainers of the system have guidance to operate the new DSS.
3. Create feedback during development and recognize that ongoing collaboration is important. Intended users, "the customers", should have realistic expectations and a at least an implicit "contract" with the developers. The eight steps provide incremental commitment of resources.
4. Provide an open, adaptive development process. Decision support needs and technologies can, and often do, change during a project. The development method must be able to adapt to significant changes in needs or capabilities. Project managers need to ensure responsive, need-based development.

In many situations a full-scale SDLC approach is too rigid for building Decision Support Systems, especially those DSS whose requirements are changing rapidly. User requirements, agreed upon at the first stage of the process, are specified with SDLC and this is often difficult to do. Any significant change in needs restarts the entire SDLC. Changes in requirements are therefore often expensive; SDLC does limit change rather than accommodating it.

Developers and future users need predictability **AND** adaptability when implementing a traditional data-driven DSS. We don't need to choose between agile and structured. We can have both.

References

Agile Introduction For Dummies

<http://agileintro.wordpress.com/2008/01/04/waterfall-vs-agile-methodology/>

Barton, P., "The George Washington University Data-Driven Decision Support Project", 2003,
Page 3/4

(c) 2019 Daniel J. Power, Power Enterprises <power@dssresources.com>

URL: <http://dssresources.com/faq/index.php?action=artikel&cat=&id=192&artlang=en>

: What method is best for building an enterprise-wide, data-driven DSS?

posted at DSSResources.COM August 15, 2003.

Inmon, W. "A Data Warehouse Development Methodology,"
<http://www.inmoncif.com/home/>

Inmon, W. "Building the Data Warehouse" (3rd Edition), Wiley, March 15, 2002.

Power, D. J., Decision Support Systems: Concepts and Resources for Managers, Westport, CT: Greenwood/Quorum Books, 2002.

Stevens, A., "Implementing the Redland Genstar Data Mart", posted at DSSResources.COM July 2, 2004.

What is systems development life cycle (SDLC)?

http://searchsoftwarequality.techtarget.com/sDefinition/0,,sid92_gci755068,00.html

Author: Daniel Power

Last update: 2009-08-31 08:18