
CDSI's software development,
testing and quality assurance
processes

*Craig Dilks and Jeanine Hammar
Clinical DataFax Systems Inc.*

What we will be talking about

CDSI's implementation of IEEE
standards for management of the
Software Development Life Cycle

OR ...

An inside look at what we do and the
tools that we use

Why are we talking about this?

- Positive feedback from DFUG audit committee
- DataFax users may learn from our processes

The Big Picture

Software Development Life Cycle (SDLC) is a series of phases:

- Problem Definition
- Requirement, Analysis and Specification
- System Design
- Implementation
- Testing
- Integration
- Maintenance

Where we are today

- DataFax 3.6 is by definition in a post-development phase
- Bug fixes, improvements, additions, new features, that are post-3.6, start the cycle over again as DataFax 3.7

What went into DataFax 3.7?

- New Features
- Customer Requests
- Bug Fixes

How do we keep track?

- Implementation of a Change Request (CR) database
- Each proposed change is fully documented (even the proposals that don't make it into the next version)
- Documentation mirrors the Software Development Life Cycle (SDLC)

CR Database



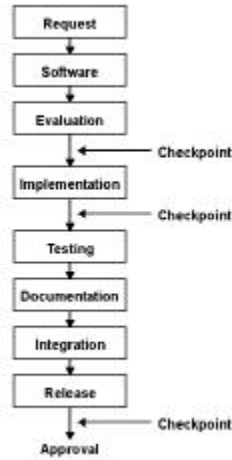
SDLC and CDSI Phases

SDLC	CDSI
Problem definition	
User requirements	User requests
Product requirements	New features, bug fixes
System design	Software design
	Evaluation
Implementation	Implementation
Testing	Testing
	User documentation
Integration and acceptance	Integration and release

How do we keep track?

- Each request, feature, bug fix is classified and assigned a unique identifier
- Cross-references link related CRs
- May be linked to a customer PR
- Approvals are required for each phase
 - all changes require re-approval
- Checkpoints at Evaluation, Implementation, and Release phases

CR Tracking



Changes to Release 3.6

3.6 Change Requests

3.6 Change Requests

Table 1. Status Summary by CR

CR	Request	Software	Evaluation	Implementation	Testing	Documentation	Integration	Release
[E-200303032] [De-032]	P WT 2003/03/21	P EB 2003/07/15	F EB 2003/07/15	F EB 2003/12/03	P []	P	P	F
	A EB 2003/04/09	A WT 2003/07/16	A WT 2003/07/16	A WT 2003/12/03	A	A	A	A
[E-200303032] [De-12]	P WT 2003/03/21	P EB 2003/11/14	F EB 2003/11/14	F EB 2003/12/31	P CD 2004/03/07	P	P	F
	A EB 2003/04/09	A WT 2003/11/22	A WT 2003/11/22	A WT 2003/12/31	A	A	A	A
[E-200303032] [De-032]	F EB 2003/04/03	F EB 2003/04/27	F EB 2003/04/27	F EB 2003/05/23	P JH 2003/11/21	P JH 2003/12/15	P	P
	A WT 2003/04/28	A WT 2003/04/28	A WT 2003/04/28	A WT 2003/05/12	A	A	A	A

An Example: CR032

Chapter 32. Change request CR36000n032

**Chapter 32. Change request
CR36000n032**

Synopsis: Using **DReportType** it is now possible to export data records in the Comma Separated Variables (CSV) format.
 Related to: None.
 Classification: sw-add

Table 32.1. Approvals

Task	Performed by	Approved by	Comments
Request	Eric Boech 2003/04/23	Wayne Taylor DVM 2003/04/28	
Software	Eric Boech 2003/04/27	Wayne Taylor DVM 2003/04/28	
Evaluation	Eric Boech 2003/04/27	Wayne Taylor DVM 2003/04/28	
Implementation	Eric Boech 2003/05/23	Wayne Taylor DVM 2003/06/13	
Testing	Jeanine Hammar 2003/11/21	Eric Boech DVM 2004/02/16	
Documentation	Jeanine Hammar 2003/12/15	Eric Boech DVM 2004/02/16	
Integration		DVM	
Release		DAM	

How do these ideas and observations become software?

- In the documentation, a Customer Request is restated
 - “The Comma Separated Variables format is a convenient and widely understood format for sharing data records. For example, Microsoft Excel easily imports data records that follow the CSV format. Would it be possible to include this output format in DataFax's export capabilities?”

Software Design

- Request translated into “software” terms

“The record writing algorithms in **DFexport.rpc** could be modified to optionally include this CSV format.”

Software Design

- Merits of the proposed addition are discussed

“The CSV format for data records is well-defined and easy to implement. The requirements of CSV are ...”

Evaluation

- A plan for solving the problem is put forward.

“The CSV format is widely used and hence is desirable as a format to support. The necessary changes can be limited to the record/field writing code. The CSV format should be an option and, for backwards compatibility, turned off by default.”

Implementation

- Plan is implemented by the developers

“A new command-line option, -z, has been added to the list of options available to the user. Use of this option indicates that the records should be written in CSV format and this is recorded internally by setting the csv variable.” ...

Implementing Code Changes

Table 32.2. Commits

Module	Rev	Diff from previous rev
datafax/src/rpc/DFexport.rpc.c	1.65	1.65 -> 1.65

Logging Code Changes



Powered by **APACHE**

CVS log for datafax/src/rpc/DFexport.rpc.c

Up to [Development] / datafax / src / rpc

[Request diff between arbitrary revisions](#)

Revision **1.65** / [download](#) - [annotate](#) - [\[select for diff\]](#) , Fri May 9 04:03:26 2003 UTC (9 months ago) by *eric*
 Changes since 1.65: +109 -25 lines
 Diff to previous 1.65

```
<code> log='all in'>src/rpc/DFexport.rpc.c</code>
<DBRC>Implementation changes to support CSV as an export format.
Source changes and developer testing completed.
These changes are in support of OR 032.</DBRC>
```


Test Plan

32.5. Testing

Test plans must test all of the requirements for CSV as described in the software section above.

32.5.1. Test Report CR36000t032

A new command line option, `-z`, allows `DFexport.rpc` to export data in CSV format. By default, CSV format is turned off unless the user specifies the `-z` option.

Testing must verify that all CSV requirements are met prior to the data being exported.

Testing has completed successfully on `sparc_solaris` and `s86_solaris` operating systems.

Number	Status	OS	Inputs	Outputs	Problems
01	pass	sparc_solaris,s86_solaris	<p>Obtain a table for testing. Create a table (table 1) that contains very few fields. In <code>DFtabledata</code>, enter only a few fields on the test table.</p> <p>Open a terminal window and enter <code>DFexport.rpc export</code>. The referred data in the output (non-CSV format). The following command should be used to export data to standard output: <code>DFexport.rpc -z gettable 001 1</code>. Run the <code>DFexport.rpc</code> command again, such that the output is written to a specified file. Verify that each exported record is one line in length after each export.</p> <p>Re-execute <code>DFexport.rpc</code> using the <code>-z</code> option to export the same data in CSV format. Use the following to export the data to standard output: <code>DFexport.rpc -z gettable -z 001 1</code>. Run <code>DFexport.rpc</code> over again that the output is written to a specified file. Verify that each exported record is one line in length after each export.</p> <p>Compare the CSV output to the default output and verify that the actual data exported is same in both cases.</p>	<p>When data is exported in non-CSV format, each exported record should be one line in length and fields should be delimited by a <code> </code>.</p> <p>When exported in CSV format, each exported record should also be one line in length and fields should be delimited by a <code>,</code> (comma).</p> <p>The exported data itself is the same in both outputs.</p>	

Documentation

- Documentation is modified to include a description of the new behaviour and how that behaviour is controlled

“The section in the DataFax Programmer Guide describing `DFexport.rpc`, was updated to include the new `-z` option and a description of its behaviour. This update also includes an example.”

Implementing Documentation Changes

Table 32.3. Commits

Module	Rev	Diff from previous rev
help/progman/xml/shell/DFexport.xml	1.22	1.21 -> 1.22

Logging Documentation Changes

Generated by
APACHE

CVS log for help/progman/xml/shell/DFexport.xml

Up to [Development] / help / progman / xml / shell

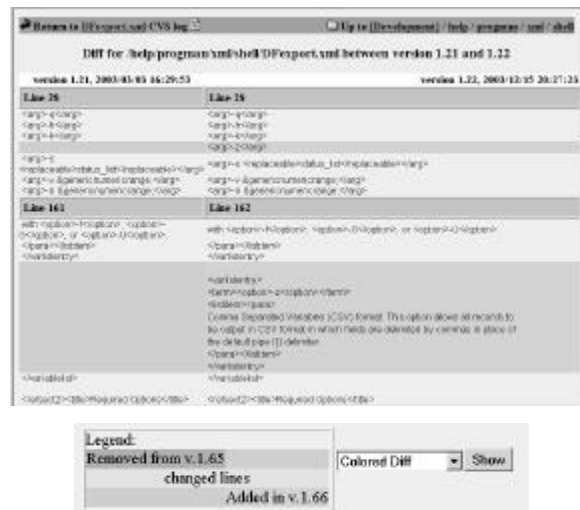
Request diff between arbitrary revisions

Default branch: 1

Revision 1.22 ([download](#)) - [annotate](#) - [\[select for diff\]](#) , *Mon Dec 15 20:27:29 2003 UTC* (8 weeks, 1 day ago) by joatise
 CVS Tags: **HEAD**
 Changes since 1.21: +70 -0 lines
 Diff to previous 1.21

```
<NAME loc="doc/progman"></NAME>
<DESC>
The section describing DFexport.rpc was updated to include the new -z option
used to output data in CSV format. An example was also added to the Examples
section of the DFexport.rpc documentation.
This change is in support of CR32.
</DESC>
```

Documentation Comparison



Integration and Release

Integration testing

- clean install of new release
- ATK
- ensure that old behaviour has not changed
- what other pieces might be affected?

Documentation tools used at CDSI

- Software Development Life Cycle implemented via an XML DTD

XML tags provide a mechanism for identifying what parts of the documentation set pertain to which phase, including the status of a given phase and any approvals that have occurred.

XML provides opportunities for automating much of the process using standard tools.

Documentation tools used at CDSI

- XML for information structure and storage
- XSLT for formatting
- HTML and Apache for delivery
- Good old vi text editor for editing
- Java-based XML/XSLT to HTML transformation processing (any number of tools to do this are available)
- CVS for decentralized content creation and centralized content management
- cron for regular documentation builds

Development tools used at CDSI

- gcc, GNU compiler
- C/C++ source code
- CVS for source code
- cron for nightly builds
- public libraries for TIFF, Ghostscript, HylaFAX
- commercial libraries for Qt

Questions?
