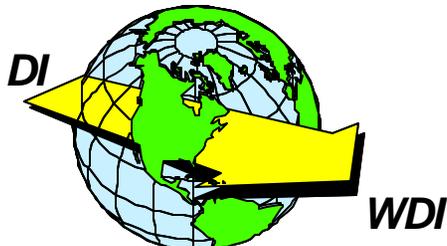


"To perform Quality Processes or not to perform Quality Processes for mapping there should be no question"

Tania Casperson
IBM Global Services

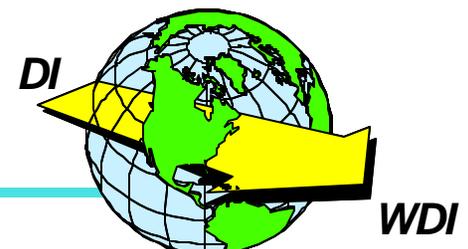
The Next Generation



2002 User
Conference

Agenda

- ★ Why we felt we needed to do this?
- ★ What is a Quality Process?
- ★ Benefits of Quality Assurance, Quality Control, Version Control
- ★ Benefits of Quality Assurance, Quality Control, Version Control
- ★ What tools /processes do we use that help us to achieve our goals
- ★ Considerations
- ★ Summary



What we are...

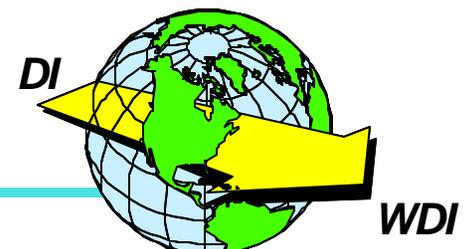
We are an organization within IBM Global Services whose mission is to be a Service provider for EDI, we provide Mapping Services, Consulting Services as well as the capability of hosting maps in our production environment.

Our customer base includes .. small(1 to 2 maps) to large customers that have outsourced all of their EDI needs to us.



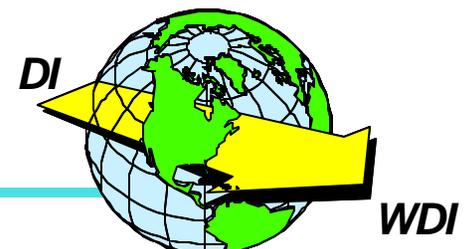
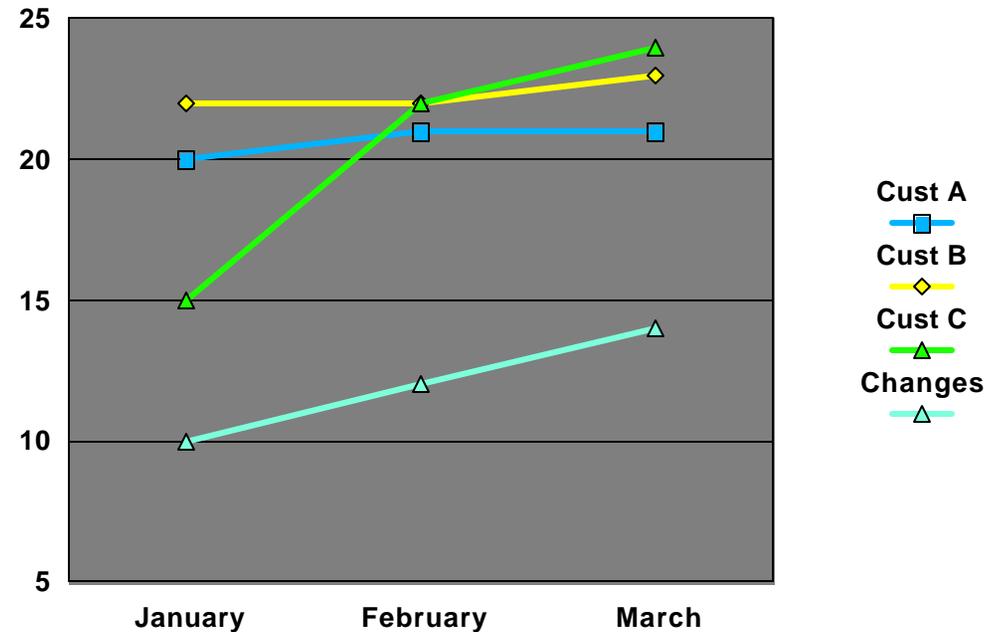
The way we were....

- ★ We had very little in the way of Version Control,
- ★ Quality Assurance or real Quality Control.
 - Sure we unit tested our maps.
 - But the story went like this... everyone did things differently.
 - ✎ With each new change, each staff member had to start from scratch.
 - ✎ No one knew what the last change was, the folder was often misplaced.
 - ✎ We relied on our hero's to come in and save the day.



What we saw....

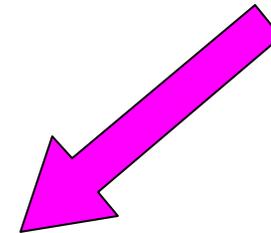
- ★ Highest Rework costs
- ★ Significant problem ticket volumes
- ★ Lower Customer Sat



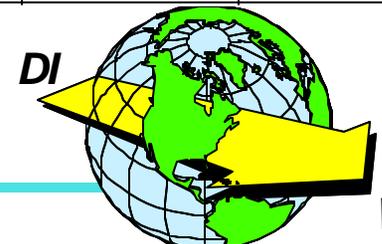
What we saw.. cont.

★ High abend counts/Support Costs:

- No version control
- No Code/project reviews
- No historical Documentation
- No support Documentation



	Nov 2000	Dec 2000	Jan 2001	Feb 2001	Nov 2001	Dec2001	Jan 2002	Feb 2002
Job Runs	41000	39000	42000	43000	48000	40000	42000	39000
Abends	125	127	88	78	10	10	3	4



What we did...

- ★ We added some controls to our process
- ★ We created entry criteria for our projects
- ★ We created exit criteria for our projects
- ★ We created Quality Assurance forms that checked that we did what we said we were going to do -Verification Checklist
- ★ We established base Unit test plans
- ★ We created a Installation Instruction form that assisted us in configuration management

Unit Test Plan/Report	<i>e-Technology Center</i>	
<Release/Component Name/Line Item Name>	Date Held:	<mm/dd/yyyy>

Unit Test Checklist

Here is a checklist to be used during unit test. I recommend that you fill out a sheet for each module/screen and file it in an appropriate place in the Project Teamroom or as an attachment to the Unit Test Case.

For each module / screen:

Has the module/screen been tested against target platform?

Has each line of code been tested with at least one test scenario?

Has each line of the design document been tested for accuracy?

Have all common errors been tested? That is, have you tested for each error message?

Have missing fields in the ini files been tested? Wrong data in the ini files? Correct data in the ini files?

Have you tested with incorrect setup (app server down, no ini file, etc.) to ensure a user friendly error message displays.

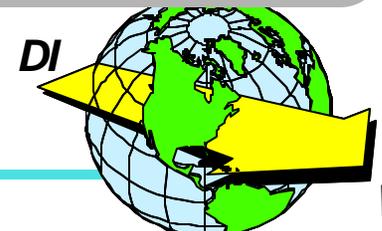
Have all simple boundaries been tested--maximum, minimum, and off-by-one boundaries?

Have compound boundaries been tested, that is, combinations of input data that might result in a computed variable that's
too small or too large?

Have you checked for the wrong kind of data, for example, a negative number for available quantity for example?

Are representative, middle-of-the-road values tested?

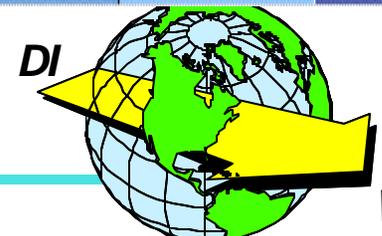
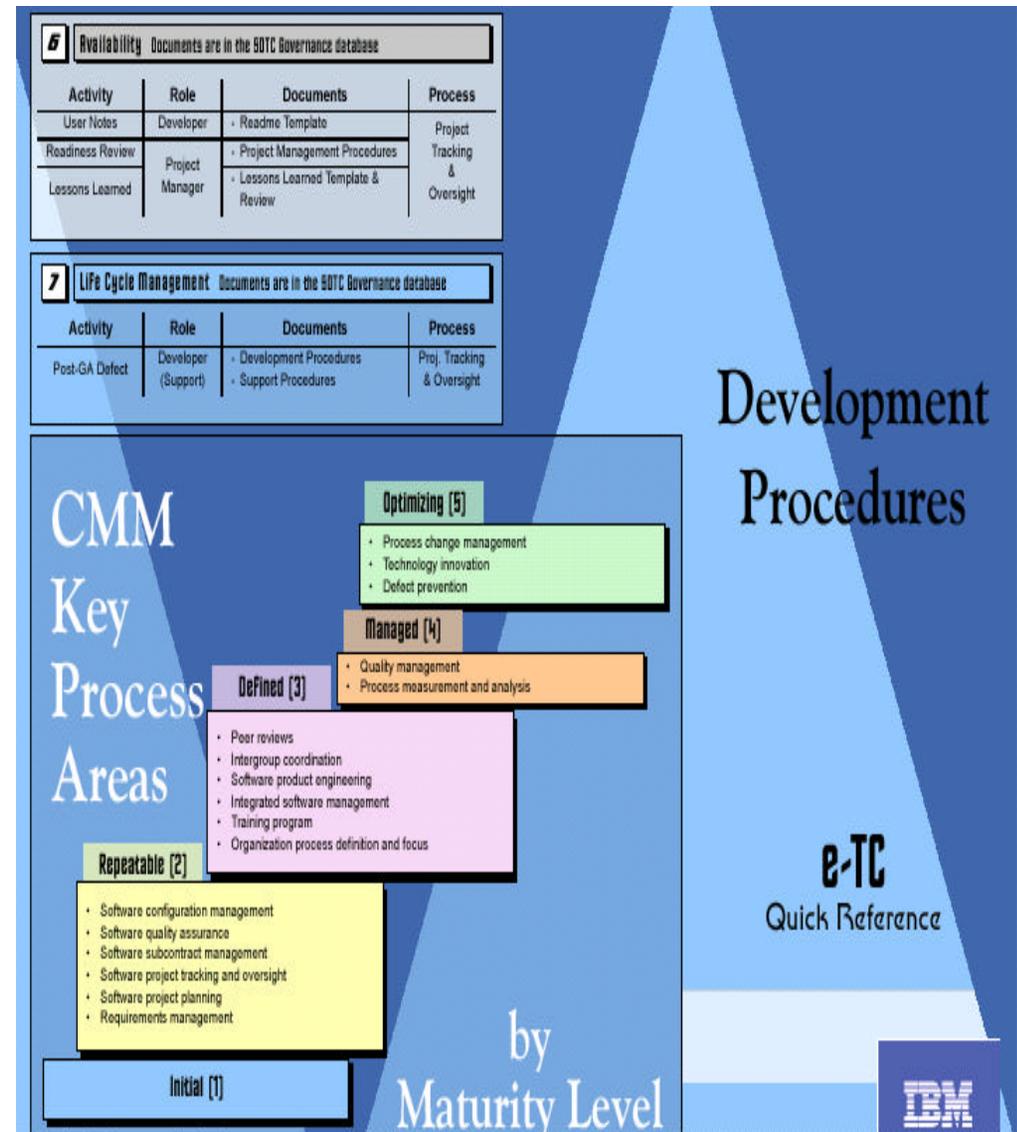
Is compatibility with old data tested?



What are the Benefits of Process Improvement?

Process Improvement

- Cuts Costs
- Eliminates Rework
- Improves Schedule Adherence
- Reduces Ongoing Support Issues
- Enhances Customer Satisfaction
- Manages Requirements Better
- Become more Competitive



How we use Version Control in our Maps

Block Copy Delete mapping Insert Map Repeat

TP13 Data Element Mappings - BIG0001 1 to 5 of 5

Transaction ID : R811ALLAZX3050

Standard transaction ID : 811

Loop ID :

Segment ID : BIG

Element ID : 373

Ref

Action Num Description

- 1 **Literal = &SET DIVARTRACE 1**
- 2 **Literal = &SET LVMAPID DI3FRMD1|R811ALLAZX3050|ALLST811AZ-I**
- 3 **Literal = &SET DIVARTRACE 0**
- 4 **Literal = &SET DIEXPTRACE 0**
- 5 **Literal = &SAVE LVPROCDATE**

★ **First we added our map Names, DI copy, and ADF names to our report.**



How we use Version Control in our Maps cont.

TP13 Data Element Mappings - BGM0002

1 to 8 of 13

Transaction ID : ABBXIORDERS96A

Standard transaction ID : ORDERS

Loop ID :

Segment ID : BGM

Element ID : 1001

Ref

Action Num Description

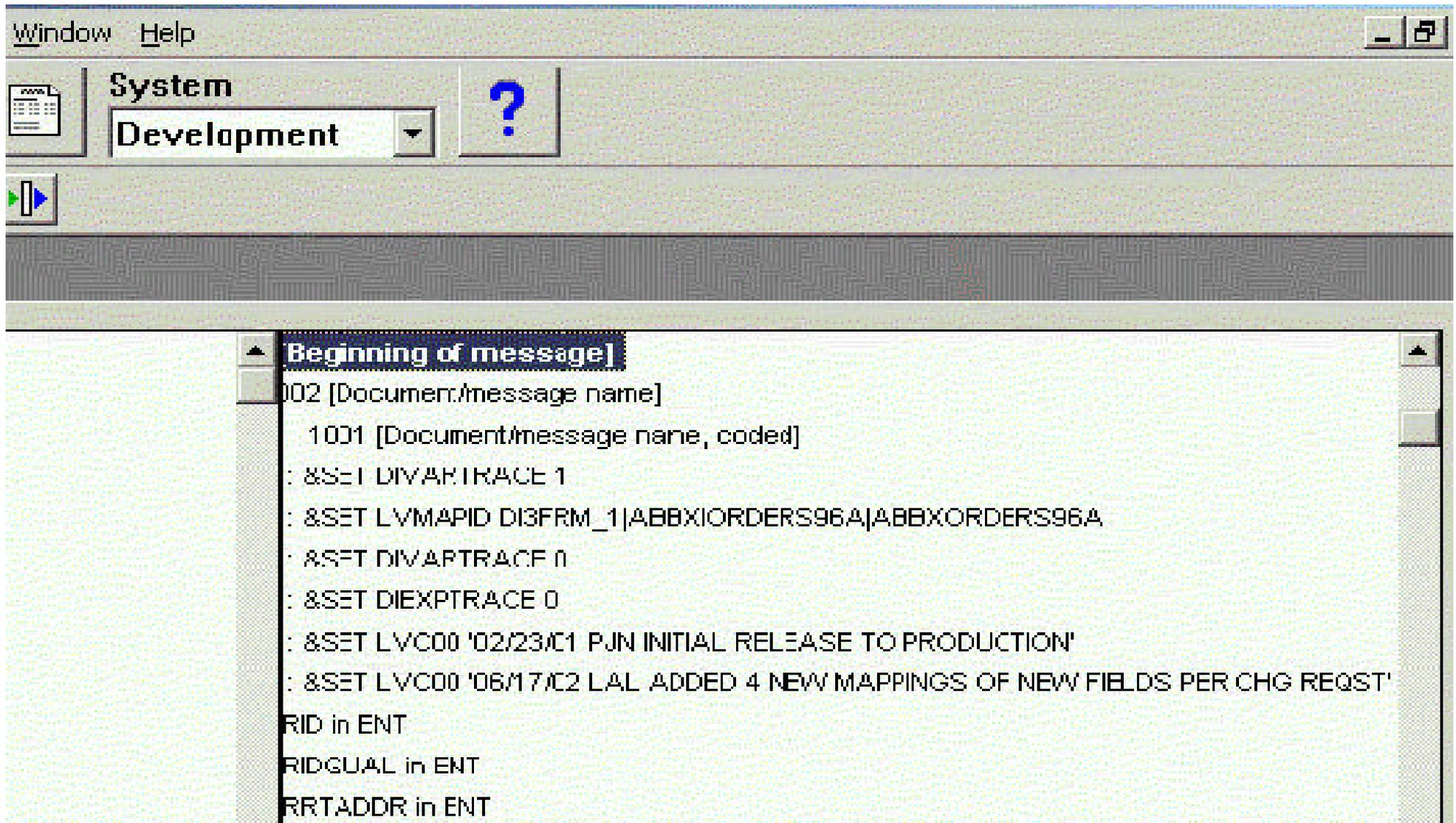
- 1 **Literal = &SET DIVARTRACE 1**
- 2 **Literal = &SET LVMAPIID DI3FRM_1|ABBXIORDERS96A|ABBXORDERS96A**
- 3 **Literal = &SET DIVARTRACE 0**
- 4 **Literal = &SET DIEXPTRACE 0**
- 5 **Literal = &SET LVC00 '02/23/01 PJN INITIAL RELEASE TO PRODUCTI**
- 6 **Literal = &SET LVC00 '06/17/02 LAL ADDED 4 NEW MAPPINGS OF NEW**
- 7 Application field name = ENT.SNDRID
- 8 Application field name = ENT.SNDRIDQUAL

★ Then when we have a change we add the date text of the change, to map to print in the report.

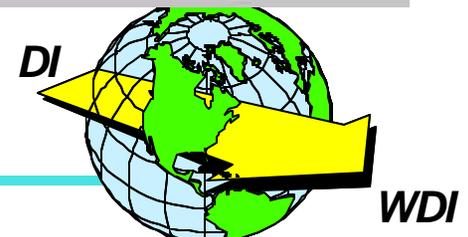
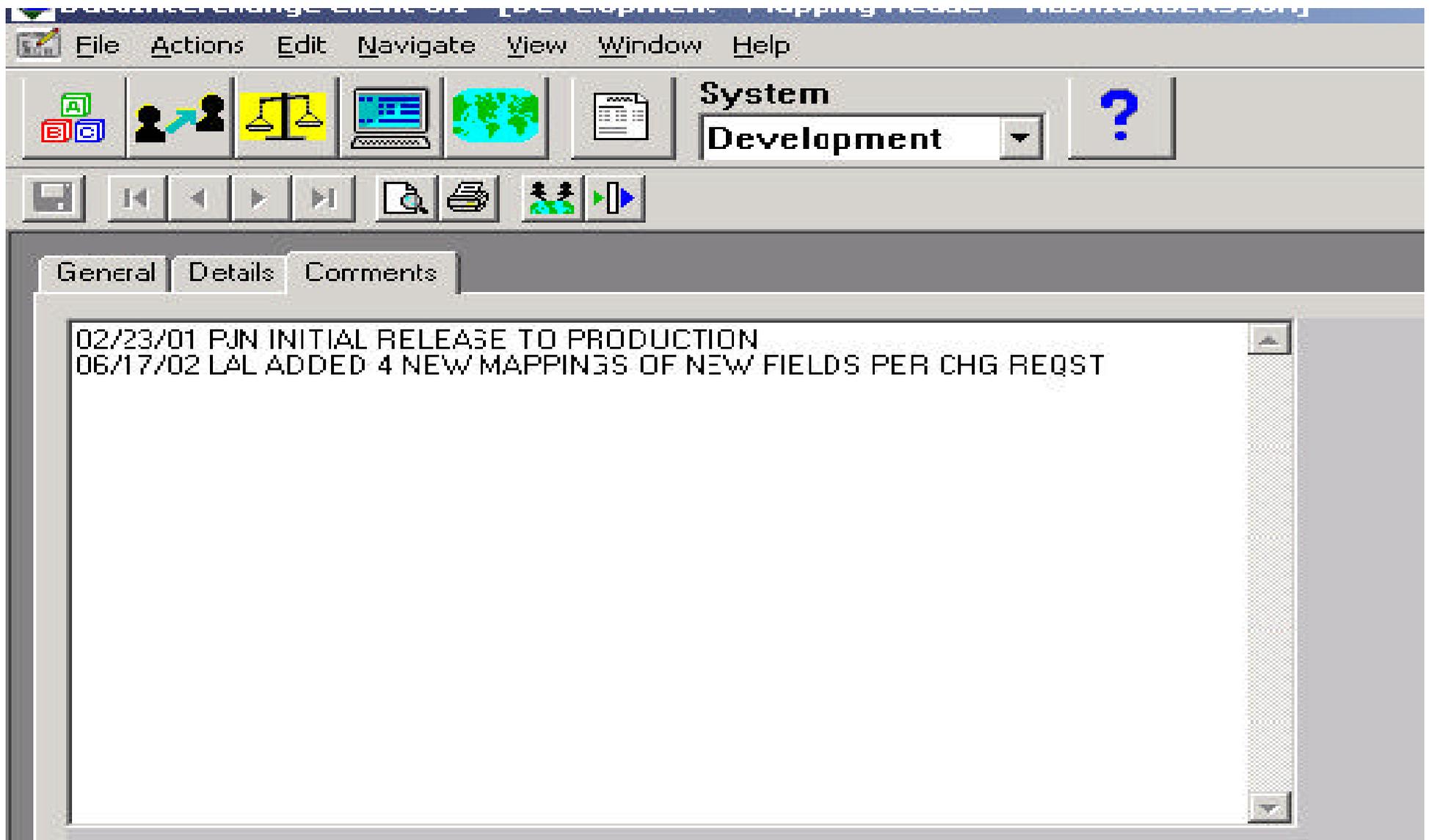


WDI

Version Control in the Client



Version Control in the Client



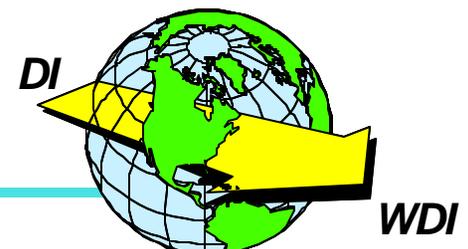
Why we use Version Control in our Maps

Benefits:

In our environment, we have multiple copies of DI as well as multiple customer's ...

In a production environment we can easily identify which map is being run, as well as understanding the last change that has occurred.

- This reduces time in researching problems.
- Ensures that the correct version of the map is installed.
- Ensures that the correct version of the map is being pulled for update.

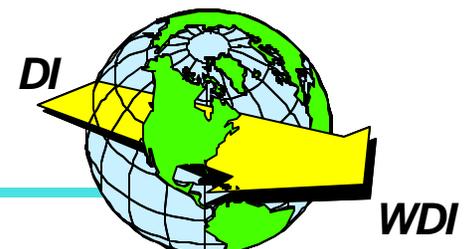


Mapping Code Reviews:

Code reviews help to establish:

- Consistency
- Eliminate errors
- Check for design flaws
- Helps to train less experienced employees

Error Severity	Error Type	Line/Page#	Comment
Minor	Standards	1	Change the D in the Di copy to X.
Minor	Design	2	Add error logic to put out a warning message in the event that a wrong code is sent. Add to documentation



Unit Test Plans and Reports

We established basic Unit Test Plans

Which contained some standard tests:

- Tested Empty Files
- Missing Enveloping Segments
- Forcing some Logic errors
- Create conditions that would force expected errors
- Ensure that the settings on the map were set correctly

Unit Test Checklist

Here is a checklist to be used during unit test. I recommend that you fill out a sheet for each module/screen and file it in an appropriate place in the Project Teamroom or as an attachment to the Unit Test Case.

For each module / screen:

Has the module/screen been tested against target platform?

Has each line of code been tested with at least one test scenario?

Has each line of the design document been tested for accuracy?

Have all common errors been tested? That is, have you tested for each error message?

Have missing fields in the ini files been tested? Wrong data in the ini files? Correct data in the ini files?

Have you tested with incorrect setup (app server down, no ini file, etc.) to ensure a user friendly error message displays.

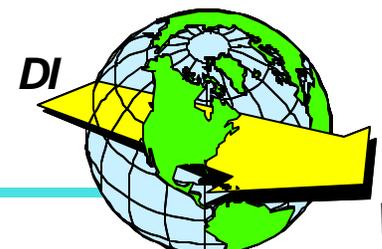
Have all simple boundaries been tested--maximum, minimum, and off-by-one boundaries?

Have compound boundaries been tested, that is, combinations of input data that might result in a computed variable that's
too small or too large?

Have you checked for the wrong kind of data, for example, a negative number for available quantity for example?

Are representative, middle-of-the-road values tested?

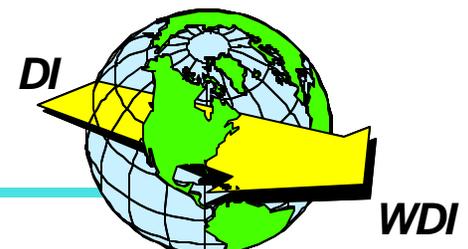
Is compatibility with old data tested?



Tools that we use :

We developed a tool that assists us in verifying our data content

We created a program in C that will take the export of the ADF layout and the resulting ADF file and create a report (like file aid will do) that will list the Field name, position, length, and then the value that is at that position.



Tools that we use :

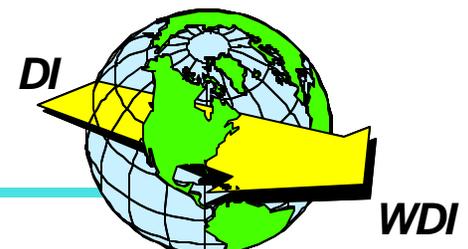
DD:REPORT - This file is a report that shows each record type and id, along with each field in the record type, the fieldname, size, position and type (according to the dd:LAYOUT file), and the contents of each field (according to the dd:ADFFILE).The program will abend if this file cannot be opened. A partial sample report is shown below for reference:

Record Type: RECORD10
ID: 10

Field Name: R10.RECID
Size: 2
Pos: 1
Type: CH
Contents: >10<

Field Name: R10.SUPLEDIMBID
Size: 10
Pos: 3
Type: CH
Contents: >1234 <

DD:ADFFILE - This file is an ADF data file that has been produced with the dd:LAYOUT layout and a customer map. The program will abend if this file cannot be opened.



Tools that we use :

Sample JCL:

```
//PGMGO EXEC PGM=EUADFCMP
//STEPLIB DD DSN=DINT.DEVL.LOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//LAYOUT DD DSN=JOANSPE.ADFUTIL.ADFTAGD,DISP=SHR
//ADFFILE DD DSN=JOANSPE.ADFUTIL.ADFOUT,DISP=SHR
//REPORT DD DSN=JOANSPE.ADFUTIL.ADFRPT,
//      DISP=(,CATLG,DELETE),
//      DCB=(LRECL=80,BLKSIZE=0,RECFM=FB),
//      UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE)
```

Inputs:

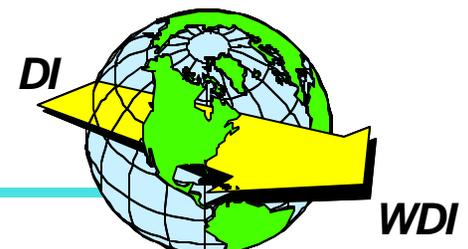
DD:LAYOUT - This file is an ADF layout that has been previously exported from DI. It MUST be exported in a tagged format, not a fixed format. The program will abend if this file cannot be opened. A partial sample of a tagged exported layout is below for reference:

```
0C1JOANSPE 010319165841ENU010301000000
2F1FORMAT(RBRN8504010) DESC(RUBBERNET 850-4010 : 03/07/01 PJN) BASESTR(POBASE)
DATE(20010317) 2F2FORMAT(RBRN8504010) STRID(POBASE) SEQNO(000) STRTYPE(SH) MAXUSE(1)
```



Tools that we use :

- ★ This assisted us in verifying our ADF data more quickly and efficiently.
- ★ Was extremely helpful when implementing the XML PTF.... made the file easier to read.
- ★ Eliminated the need to create a Cobol Layout, we could use the export of the ADF in DI directly... saved us lots of time.



Tools that we use:

Verification Checklists:

Lists all of the tasks , settings, deliverables ... essentially all of the details of the project as well as some common problems noted in previous Installation Reports(lessons learned).

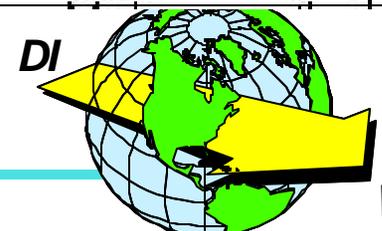
We have a different developer review another's assignment and use the verification form to ensure nothing is missed..... this helped in reducing installation problems.



Tools that we use :

Verification Checklist

OK	N/A	Item
		Project Information
		- All items completed
		- Completed sections of form are indicated in index table
		- Verifier has confirmed that all PDS members listed on the form to be installed do exist in their appropriate source library.
		- Additional testing has been performed by verifier. If it has, attach additional checklist for quality testing.
		JCL Installation Instructions
		- Appropriate fields for this section of installation form are filled in
		- Verifier has jumbed production-ready jcl in staging library, and confirmed that any errors are expected (e.g. a program not found because it has not yet been moved to production source library).
		- After the job has been jumbed, if any VSAM files need to be created, make certain these are on the installation form. Make sure the name of the library from which the contents need to be copied is accurate.
		- After the job has been jumbed, if there are any GDG's which need to be



Tools that we use :

		Map Export/Import Instructions
		- Appropriate fields for this section of installation form are filled in
		- FROM and TO copies have been verified as correct
		- Mapname and ADF filenames have been verified as correct; developer has pointed out changes made (for change requests)
		- Export/import filenames have been verified as correct, and any new trading partner profiles and usages are contained therein and have been verified as correct.
		- Verifier has confirmed that before and after testfiles have been compared, and new results are correct.
		- The install request shows whether map pieces were exported separately or together.
		- Any change has been documented in the map's modification log
		Card Installation Instructions
		- Appropriate fields for this section of installation form are filled in
		- Contents of card members have been verified as correct by viewing of actual results of existing card, if possible (e.g., view an email that has been produced but not sent out, or view a production database relationship to verify that ID on card matches the db)
		Code Installation Instructions
		- Appropriate fields for this section of installation form are filled in
		- Results of test files have been verified as correct, with visual and SUPERFC



Tools that we use:

Installation Instruction Form:

- This form lists all of the installation elements for our production support staff that does our installations.
- The form also provides any additional information that may be essential for support, or the installation of the component.
- The production support staff reviews the form and conducts meetings if necessary (when multiple maps, jcl streams are installed).
- Any noted problems are reviewed in a weekly post install meeting and all issues are noted in the verification form.
- This has essentially eliminated any install problems that we had been experiencing previously.



Tools that we use:



EDI Development Services - Installation Instructions

Project Information:

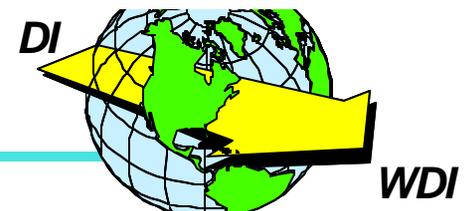
Project Name/Number:	
Project Manager:	
Project Claim Code:	
Project Description:	
New Implementation or Change Request?	
Priority:	
Target Date for Installation:	
Form Verified by:	

Installation Form Completion Instructions:

This form is intended to provide clear and accurate production installation instructions for all Data transformation Service elements.

- o Text entry areas are comprised of tables. Therefore, the <ENTER> or <TAB> key should be used when additional space/lines are needed.

Indicate completed sections of this



Tools that we use:

Indicate completed sections of this form with an "X" below:

<input type="checkbox"/>	JCL Installation Instructions
<input type="checkbox"/>	Map Export/Import Instructions
<input type="checkbox"/>	Card Installation Instructions
<input type="checkbox"/>	Code Installation Instructions
<input type="checkbox"/>	STEDI / IE Information
<input type="checkbox"/>	Master Schedule Changes / OPC Info

- ▶ JCL Installation Instructions
- ▶ Map Export/Import Instructions
- ▶ Card Installation Instructions
- ▶ Code Installation Instructions
- ▶ STEDI / IE Information
- ▶ Master Schedule Changes / OPC Info



Tools that we use:

Import/Export Instructions:

BACKUP MAP Instructions:

1. Select DI Option #4, "Trading Partner Transactions (Mapping)"
2. Check to see if the Backup Map already exists; If it exists, delete it.
3. Locate the specified Map Name and enter "C" (Copy).
4. Enter the Backup Map Name specified and set the "Migrate usages (Y/N)" field to "Y".
5. For each usage displayed, enter "I" (Migrate).

EXPORT Instructions:

1. Before logging into the DI copy you intend to export from, remove all contents (edit, delete or rename) of level qualifier (User ID). (ie: "userid.EDI.EIFILE")
2. Log into the specified copy of DI.
3. If a Backup Map Name is specified, make a copy of the map. Follow the BACKUP MAP instructions.
4. Select DI Option #10, "Export".
5. Select DI Sub-Option #3, "Trading partner transactions".
6. Locate the specified Map Name and enter "E" (Export).
7. Set all Y/N option fields to "Y".
8. After successfully exporting the map, LOG OFF the copy of DI you exported from.
9. Rename the DI export/import filename (ie: "userid.EDI.EIFILE"), to the requested export/import filename
10. Reply to this installation form to notify the requestor that you have completed the export.

IMPORT Instructions:

1. Before logging into the DI copy you intend to import to, delete the DI export/import filename under your ID ("userid.EDI.EIFILE") and copy the specified Export/Import Filename to your DI export/import filename.
2. Log into the specified copy of DI.
3. If a Backup Map Name is specified, make a copy of the map. Follow the BACKUP MAP instructions.
4. Select DI Option #10, "Export".



Summary:

- ★ As a Mapping Service provider the need for real quality and speed is paramount to a successful Service.
 - Using Process Improvement disciplines assisted us in reducing costs, enhancing our customer satisfaction, eliminating rework, becoming more competitive

These disciplines included:

- ★ Version Control
- ★ Quality Control
- ★ Configuration Management
- ★ Code Reviews
- ★ Unit Test Plans

Tools:

- ★ ADF reporting Tool
- ★ Verification Form
- ★ Install Instructions



?

?

?

Questions

?

?

