EDUC 2018

European DataFlex users' conference - Edinburgh, Scotland

DataFlex to New Heights

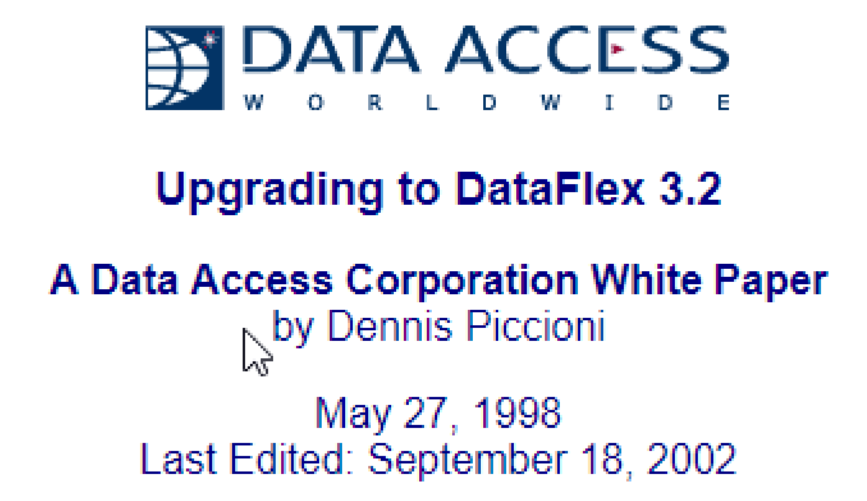
Skipping Windows

Marco Kuipers – 28 IT

Skipping Windows!

Imagine modernising a 30-year-old, 1000+ screen ERP from DataFlex 3.1d to 19.0 V , automate testing, mana ope creep.

ed.



DATA ACCESS
WORLDWIDE

Upgrading to DataFlex 3.2

A Data Access Corporation White Paper
by Dennis Piccioni

May 27, 1998
Last Edited: September 18, 2002

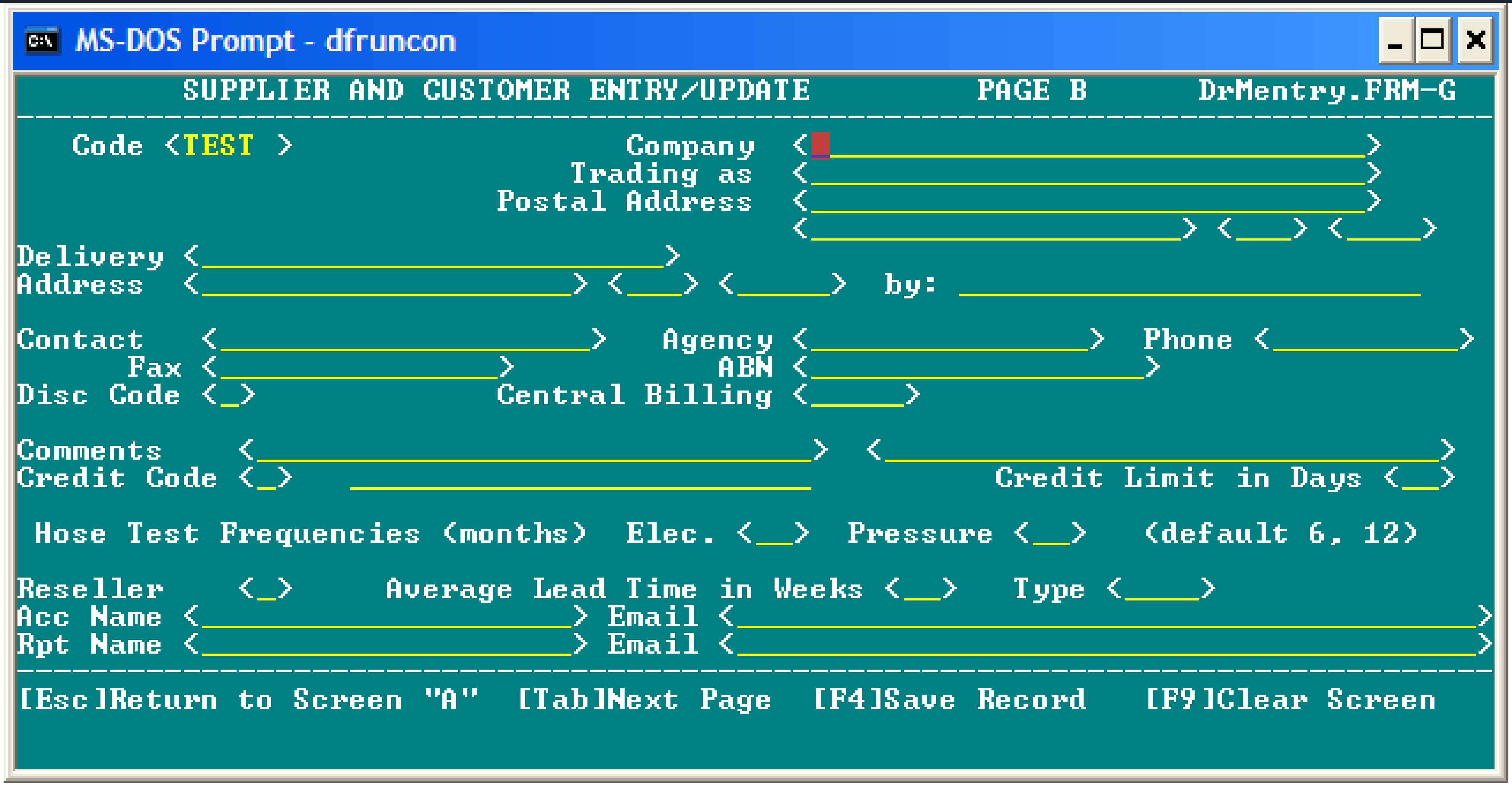
Marco Kuipers
Marco@28it.com.au

Skipping windows – Modernising software

AGENDA

- Why would you care about Character Mode customers?
- Windows, Web Desktop or Web Touch Application style
- The great things of Mobile Web
- Tips & Tricks

Skipping windows – modernising software



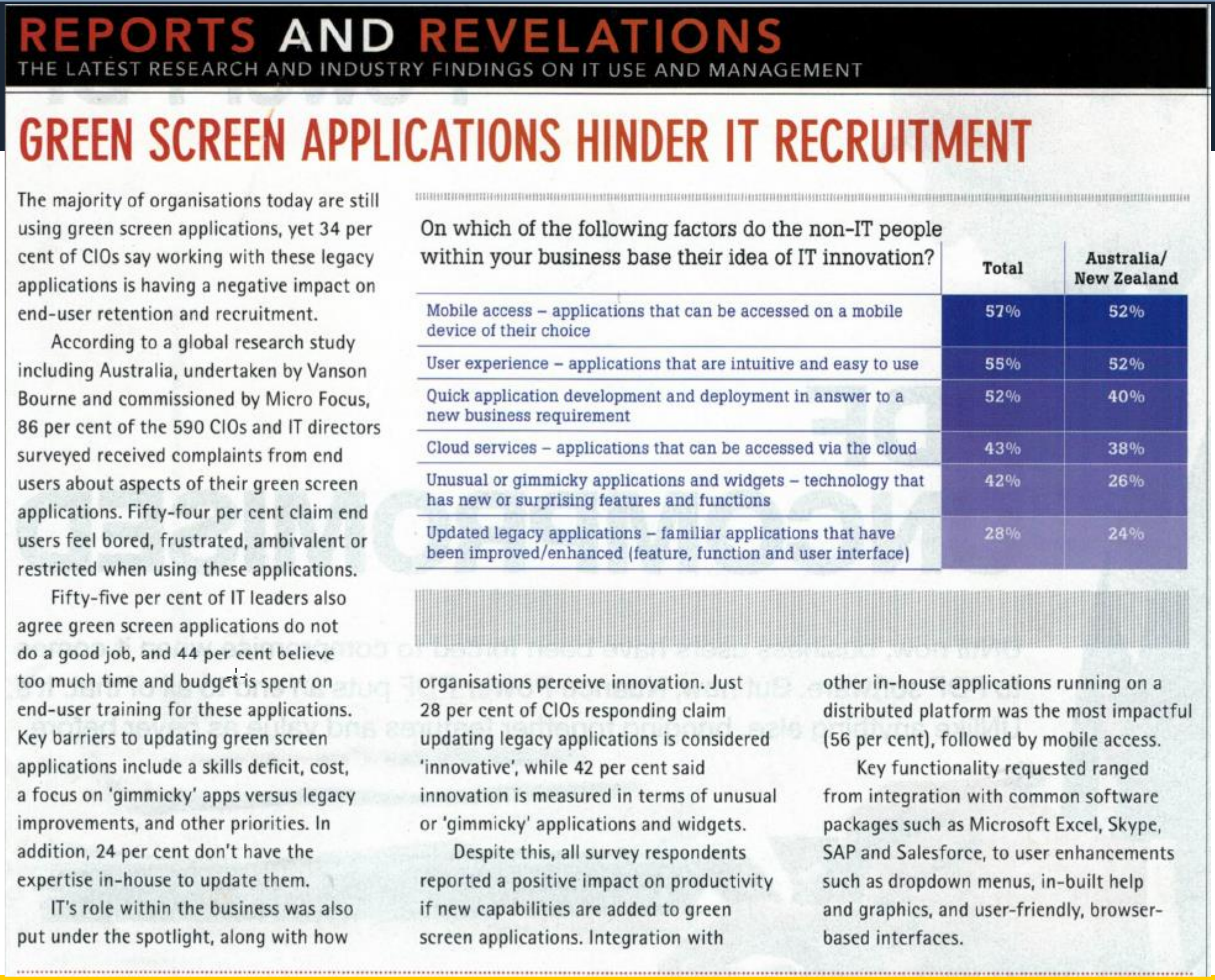
```
          SUPPLIER AND CUSTOMER ENTRY/UPDATE          PAGE B      DrMentry.FRM-G
--------------------------------------------------------------------------------
   Code <TEST >                           Company <_____________________________>
                                       Trading as <_____________________________>
                                   Postal Address <_____________________________>
                                                  <___________________> <___> <____>
Delivery <_________________________>
Address  <____________________> <___> <_____>  by: _________________________

Contact   <____________________>    Agency <_______________>  Phone <__________>
      Fax <_______________>                ABN <__________________>
Disc Code <_>                   Central Billing <_____>

Comments    <______________________________>  <______________________________>
Credit Code <_>     _________________________      Credit Limit in Days <__>

 Hose Test Frequencies (months)  Elec. <__>  Pressure <__>   (default 6, 12)

Reseller    <_>     Average Lead Time in Weeks <__>   Type <____>
Acc Name <____________________> Email <________________________________________>
Rpt Name <____________________> Email <________________________________________>
--------------------------------------------------------------------------------
[Esc]Return to Screen "A"  [Tab]Next Page  [F4]Save Record   [F9]Clear Screen
```

Common reasons not to modernise

- Too expensive
- Developer retired
- It is running fine

Discuss with the customer the dilemma

- **Affordability:** Is it ever going to become cheaper?
- **Availability:** Will one day more developers be available that understand old style DataFlex code?
- **Continuity:** What would happen to your business when the software stops working?
- **Value:** If you want to sell your business, what is it worth when its software system is in its current state?
- **Growth:** Can you attract and keep your staff with these systems into the future



REPORTS AND REVELATIONS

THE LATEST RESEARCH AND INDUSTRY FINDINGS ON IT USE AND MANAGEMENT

GREEN SCREEN APPLICATIONS HINDER IT RECRUITMENT

On which of the following factors do the non-IT people within your business base their idea of IT innovation?	Total	Australia/ New Zealand
Mobile access – applications that can be accessed on a mobile device of their choice	57%	52%
User experience – applications that are intuitive and easy to use	55%	52%
Quick application development and deployment in answer to a new business requirement	52%	40%
Cloud services – applications that can be accessed via the cloud	43%	38%
Unusual or gimmicky applications and widgets – technology that has new or surprising features and functions	42%	26%
Updated legacy applications – familiar applications that have been improved/enhanced (feature, function and user interface)	28%	24%

The majority of organisations today are still using green screen applications, yet 34 per cent of CIOs say working with these legacy applications is having a negative impact on end-user retention and recruitment.

According to a global research study including Australia, undertaken by Vanson Bourne and commissioned by Micro Focus, 86 per cent of the 590 CIOs and IT directors surveyed received complaints from end users about aspects of their green screen applications. Fifty-four per cent claim end users feel bored, frustrated, ambivalent or restricted when using these applications.

Fifty-five per cent of IT leaders also agree green screen applications do not do a good job, and 44 per cent believe too much time and budget is spent on end-user training for these applications. Key barriers to updating green screen applications include a skills deficit, cost, a focus on 'gimmicky' apps versus legacy improvements, and other priorities. In addition, 24 per cent don't have the expertise in-house to update them.

IT's role within the business was also put under the spotlight, along with how organisations perceive innovation. Just 28 per cent of CIOs responding claim updating legacy applications is considered 'innovative', while 42 per cent said innovation is measured in terms of unusual or 'gimmicky' applications and widgets.

Despite this, all survey respondents reported a positive impact on productivity if new capabilities are added to green screen applications. Integration with other in-house applications running on a distributed platform was the most impactful (56 per cent), followed by mobile access.

Key functionality requested ranged from integration with common software packages such as Microsoft Excel, Skype, SAP and Salesforce, to user enhancements such as dropdown menus, in-built help and graphics, and user-friendly, browser-based interfaces.

Majority of large organisations still using green screens.

34% of CIO's say, working with these legacy applications impact on end-user retention and recruitment.

Trendlines CIO,
Mar/Apr 2014
Pg. 62-63

Only one way forward… Modernising

Encourage the customer to reinvest in the software that has supported and grown their business in the last 20 to 30 years and modernise the software so its core functionality will last another 20 to 30 years.

Consider separation of core functionality and peripheral

Allow the customer to continue to stand out from their competitors. You can’t stand out with off the shelf software...

Once modernised the customer can

- > Grow, Continue to build the system processes around your business processes, not the other way around.
- > Access ones as well as th
- > Uptim s downtime
- > Optim ecome simple on

The skill of modernising is:
Knowing the New &
thoroughly understand the old

- > Extend Build interfaces to customers and suppliers incl. hybrid blockchain solutions.
- > Unlock Any modern developer or student can self teach DataFlex and maintain your system ONCE MODERNISED.

Modernise to what?

- CharMode to Windows, Web Desktop or Web Touch?
- Embedded Database, keep or migrate to SQL?
- Per module or whole-of-system
- Affordable refactoring (Reuse as much as possible, consider concessions you would not make for new build e.g. DD & connected grids vs offline grids & EOD run).

Why Web Touch?

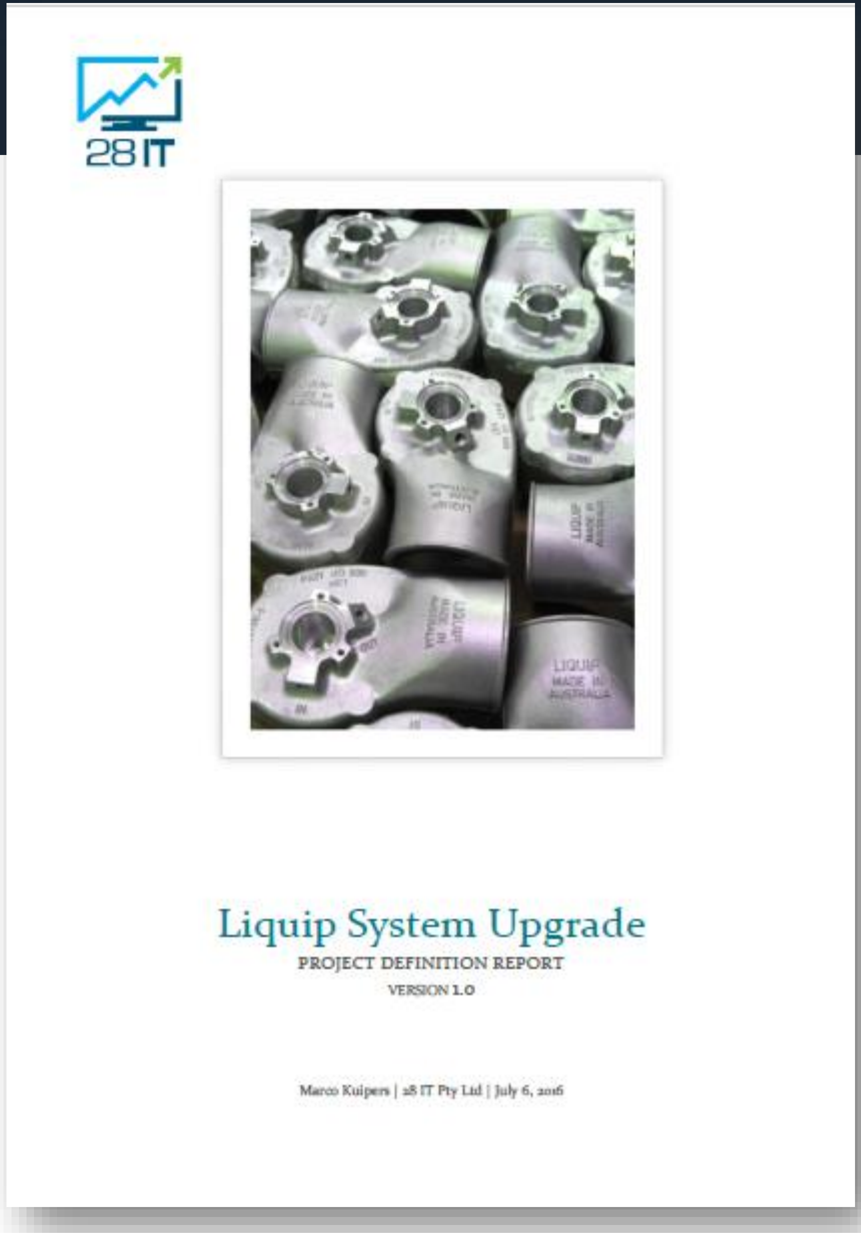
- It gives the customer a real leap, and ensures the longevity of the new system. Windows is current but web and especially touch is future.
- The amount of information on the screen is similar to the Character Mode 80x24 screen.
- The Drill down interface is akin to the Chain wait user experience

How to quote?

- This must be a paid quote
- You must have the source code
- Build a code analyser
- Clearly specify what you perceive IN scope and OUT scope
- Pretty diagrams gives confidence
- Use Sture’s DFMatrix

Project definition report

- For this report I Charged 45 hours (up front).
- In the report provide and estimation, proposed timeline.
- In this report, I worked on 50 (2.3 style) code lines per hour and timeline based on one day a week.
- The one day a week is good for both parties. For me it allows me to schedule in parallel to other clients, for the customer so they can keep up with testing.



28IT

Liquip System Upgrade

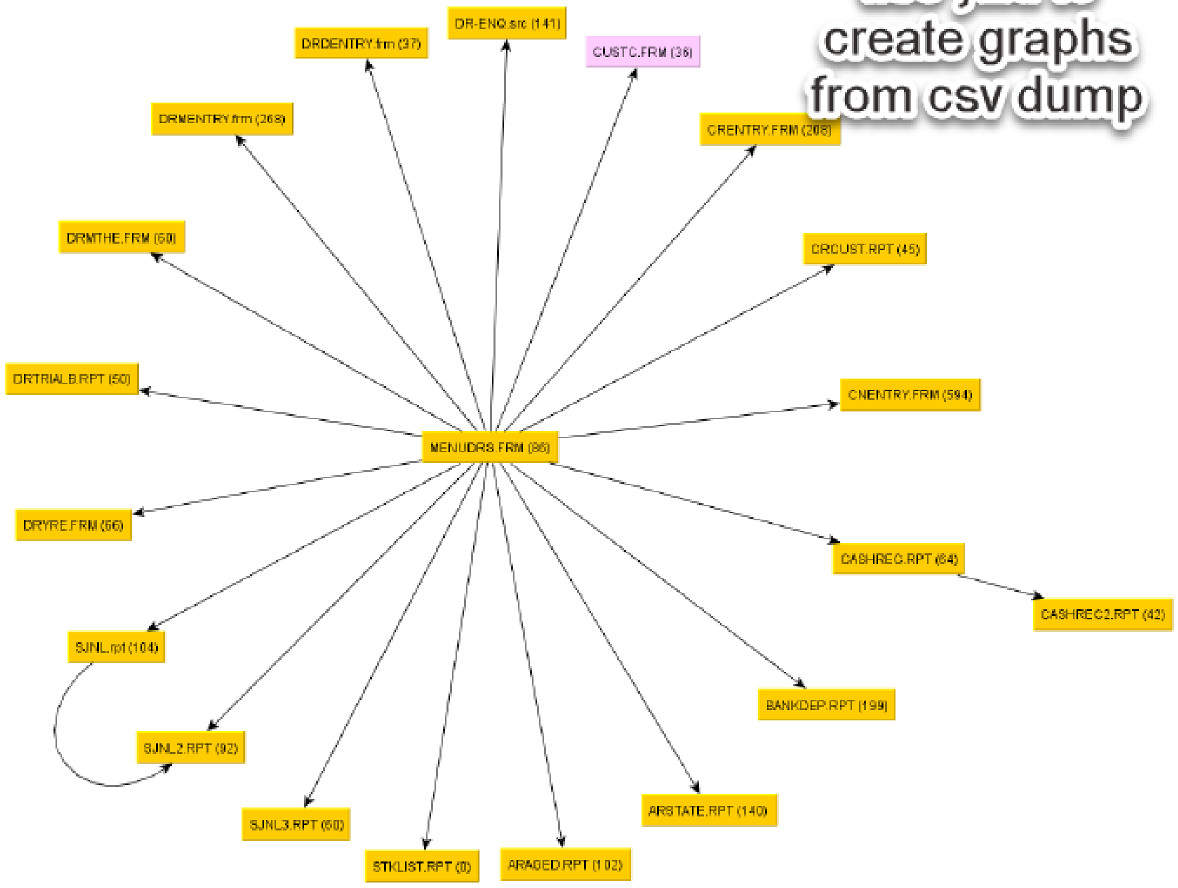
PROJECT DEFINITION REPORT

VERSION 1.0

Marco Kuipers | 28 IT Pty Ltd | July 6, 2016

One of the 24 menu’s

DEBTORS MENU (MENUDRS)



use yEd to create graphs from csv dump

Getting the job

- Hopefully you get the job, but even if they are initially shocked, you have given them a path.
- Conversations will start and the business at least has a (renewed) relationship with a trusted partner that genuinely wants the best for their business.
- The report – because its paid separately and upfront - will allow them to get any other quotes if they like.
- Most likely they want to build on the relationship.

Starting the build

- Build the test server, with an automatic daily download of production data (4am).
- Build a transform that applies any datafixes, new tables, extra fields, fixed field types in relationships etc. that runs direct after the daily download.
- Frequent drops - communicate any drop clearly via email
- Monthly status reports on where you were planned to be and where we are, including any reasons for changes.
- I started on 27 old lines per hour, as a lot of the infrastructure and foundation had to be build, but am now tracking on 68 lines per hour (avg 45.5 since start). I'm on 73%

Code control

- Use git version control, and commit often
- Re-indent the OLD code (Use the Hammer) (also source control the old system)
- Use regular expressions to convert code quick and reliable.
  - It might look difficult but it really is not and saves you so much time (use Notepad++ for replace)
  - Remove end of line comments with just initials and dates.
  - Trim the source code
  - Build own library

Build own Tools & keyboard macro's

Build tools for any repetitive task like

> Extract / compare data

> Convert Report Images to Report Coordinates / code

Use a tool like AutoHotKey for

> Lining up 'to'

> Append line to clipboard

> Convert Excel data to RDS data

Use Web DD Debug Inspector

Menu to Zoom or Select

- Web Mobile Framework, suggests to start with a list (select) then do New or Select to the zoom.
- If the user 99% of the chance knows what the key is (they are holding the order form in their hands). Provide a zoom with prompt. This is more Desktop style.

Data Dictionaries

Specify in Data Dictionaries

- Capslock, Required, Commit, status help
- Field masks to Date, Currency
- Validate_DEOs_Only_State can help you out, for invalid not-on-screen validation issues of existing old records)
- Labels, Although just for default psLabel
- Field defaults & Validations
- Combo box values (Description validation tables

DD Field value check (for combobox)

```
// 3.x style
Entry USER.DEPT {CHECK="IT|HR"}

// Windows style but not (yet?) supported with Javascript in Web Mobile
Set Field_Value_Check Field USER.DEPT to "IT|HR“

// Best for Web Mobile
Object oUserDept_VT is a ValidationTable
    Procedure Fill_List
        Forward Send Fill_List
        Send Add_Table_Value "IT"
        Send Add_Table_Value "HR"
    End_Procedure
End_Object
Set Field_Value_Table Field USER.DEPT to oUserDept_VT
Set Field_Class_Name Field USER.DEPT to "dbComboForm"
```

DD Update/Backout – Modernisation Trap

- Realtime vs Batch
  - The old system was used to end of day processing that run for hours. E.g. all customer outstanding's were recalculated based on the new, updated and invoiced orders.
- Customer expectation like-for-like
  - For now, we kept this, as thus is what they are used to. But the end of day is now scheduled and runs within a couple of minutes. Changing would arguably be scope creep.
- Totals could become DD, or SQL views
  - When the customer decides on real-time dashboards / reports, this decision will be made.

Multi row entry (grids)

- DAW: Not recommended in Mobile Touch
- In Character mode, programmers when to great length wrestling with WindowIndex to get multi line entry to work.
- My customer is not often going to do multi line entry on a phone but will on touch screen desktops (not far away now) and tablets with keyboards.
- As like-for-like conversion means that the user can enter many rows, before they do a save: use non DataBound grids with saving on button press /navigation.

Reporting

- Reporting in CM, was Matrix printers, Carbon paper, continuous tractor feed paper, and lots of physical report, and copy invoice paper storage.
- The new world is pdf.
- All reports are created using DataFlex reports or Vpe, to pdf and stored on the server. A catalogue entry is created for each print job with:
  - Report name
  - Run date & time
  - Login name
  - Filename
  - NamedValues
  - Key (The key field is optional and the only value that is not automatically generated by (my) printing framework. The key I use for e.g. reprinting the last invoice.)
- Users with the rights for Own or All print jobs, can pickup any previous Print job and re-look at, when talking with the customer or reprint when required. We have looked into PDF stamping ‘Duplicate’ but decided as not required for golive date.

Like – for – like Testing

- The customer expects a like-for-like system, while
  - Temp tables to arrays of structs
  - Multi row images and enter macro’s to cWebGrids.
- Parallel run any processes, then export all modified tables to csv (create a dump tool)
- Use Beyond compare to find/view & explain/fix differences.

User navigation logging

- Build class that saves any user navigation into the table. This gives confidence that programs are being tested, at build/test stage, as well as assists support during the rollout stage and auditing when required (cNavigationLogger).

Scope creep

I HIRED A CREEP TO HELP DETERMINE OUR PRODUCT'S FEATURES.
scottadams@aol.com
www.dilbert.com
YOU NEED MORE FEATURES.
GOOD WORK.
2/5/01 © 2001 United Feature Syndicate, Inc.
WHEN CAN YOU HAVE THAT DONE?
GAAA!!

- Stick to the original project definition report
- Or invoice separately

Demo

- Show Login
- Barcode scanner own menu
- Grid entry
- Amcharts (lib by Raphael Theiler) is great, but scope creep
- Autohotkey
- Show images to positions tool

Tips and Tricks (1/2)

- Clear table after any coded find (onValidation)
- OnLoad vs onShow (e.g. aged trial balance)
- Reporting; Separate data collection from reporting (RDS) or Array of Structs
  - Much easier for testing
  - Easier to debug / allows for data dump/load for remote issues
  - Existing DF formatting functions
  - Simple to switch between Embedded & SQL
  - Fast, no odbc and path issues

Tips and Tricks (2/2)

- Domain login - https://tinyurl.com/dfsso28
- URL parameter for Barcode scanner device
- Email popup for invoices, purchase orders etc.
- Report archive (Rights restricted)
- Navigation log (demo)

What would I do different?

- Build universal DD update tool
  - Detects Decimal xxxxx.xx fields, and generates the currency mask logic in DD
  - Detects entry {check=“ABC”} and generates validation table/combo logic
  - Detects uppercase (as {CAPSLOCK} or ASCII columns in tables where random 1000 records are all uppercase
- Learn Regular Expressions earlier
  - Huge timesaver in refactoring

Conclusions

- Web Mobile Touch is a mature framework
- The navigation is very compatible with the old Chain Wait
- Encourage CM users, to skip windows and modernise now
- Touch on tablets, Laptops and soon Desktops is the norm
- Previous time consuming ‘Enter Macro to DD’ migrations are much easier now with:
  - DD Relationships
  - NavigateData struct
  - Web objects – DataBound or Manual

Thank you!

Are there any questions?

Marco@28it.com.au