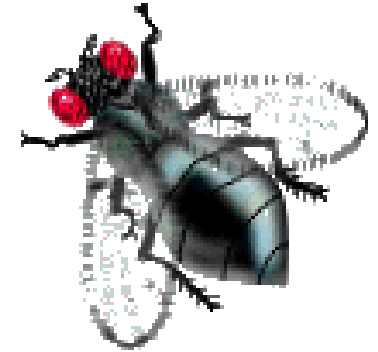




An Phríomh-Oifig Staidrimh
Central Statistics Office



CSO's Experiences testing a complex CAPI Instrument

Background



- CSO has been using Blaise since 1997 for CAPI household surveys
- In 2011 the European Central Bank commissioned the Household Finance and Consumption survey (HFCS)
- A Blaise Questionnaire was required to query household and personal assets, liabilities, wealth, income and indicators of consumption

Why a new testing approach?



- HFCS was a very complex survey Instrument
- Survey Instruments have been difficult and time consuming to test
- A new approach was needed to prioritize Questionnaire testing and also to ensure greater test coverage of the instrument



Testing in the Instrument Development lifecycle



- Requirements testing
- Component testing
- Independent Component testing
- System [& Integration] testing
- User Acceptance testing

Requirements Testing



Requirements Testing



Who ?

- Performed by the Development manager in collaboration with:
 - Specification authors
 - Programmers

Requirements Testing



Types of Tests:

- Functional or Black Box testing
- Static analysis – reviews of documentation
 - Informal reviews
 - Walkthroughs
 - Technical reviews

Component Testing



Who ?

- Component [block] programmer

Component testing



Types of Tests:

- Structural or white Box testing
- Static analysis – reviews of code
 - Informal reviews
 - Walkthroughs

Independent Component Testing



Who ?

- Anyone but the component author

Independent Component testing



Types of Tests:

















































- Black box functional testing
- Test log template for each test approach:
 - Routing
 - Variable Ranges
 - Fill/Inserts & text
 - Error/Signals
 - Computations/Don't knows & refusals

Creating test Logs




- Test Logs created from specifications
- Time consuming – worth the effort in Quality terms
- Encouraged authors to use test design techniques to create test cases

Route Tests

Household Level	Test Log Template	Completed Test Log	Error Log	Specification Document Page Number(s)	Tester	Changes Made?
Household Data				1 > 8	John	Denis
Relationships				9	John	Done
Housing Module			N/A	9 > 14	John	N/A
Tenure				14 > 20	John	Denis
Mortgages & Loans				20 > 70	John	Denis
Other Properties				70 > 118	John/Denis	N/A
Vehicles				118 > 132	Denis	Denis
Valuables				132 > 133	Denis	N/A
Financial Liabilities				133 > 150	Denis	Denis
Financial Investments				150 > 158	Denis	Denis
Inheritance				159 > 172	Caroline	Caroline
Business Interests			N/A	173 > 183	Jacqueline	N/A
Household Consumption			N/A	183 > 184	Jacqueline	N/A
Attitudes about Saving			N/A	185 > 197	Jacqueline	N/A
Interviewer Questionnaire (Paradata)				1 > 5 of separate spec	Denis	N/A
Self Completion Tests					John	To Do
Self Completion changes						Conor
HFCSSurvMgmt changes						Conor
BankAcc change						Conor

HFCS 2012: Vehicles - Routing test log template

<= Less than or equal to
 <> Not equal to
 >= Greater than or equal to
 [i] Any number between 1 and 20

Higher Level Condition	Variable Name	Page Ref	Test Performed	Expected Result	Actual Result	Fail / Pass
	Cars	123	Cars = 1	Cars_No	Cars_No	P
			Cars = 2	Oth_Veh	Oth_Veh	P
Cars_No	124	Cars_No = 0	Oth_Veh		Error message	Fail
		Cars_No = 1 to 4		Cars_Val	Fail	
		Cars_No > 4	CarsVal	Signal , followed by Cars_Val	P	
Cars_Val	124	Cars_Val <> EMPTY	Oth_Veh	Oth_Veh	P	
Oth_Veh	124	Oth_Veh = 1	Oth_VehT	Oth_VehT	P	
		Oth_Veh = 2 AND Cars = 1	Veh_Buy	Veh_Buy	P	
		Oth_Veh = 2 AND Cars = 2	Valuables > 	Valuables	P	
		Oth_VehT = 1	HowMany (MotBk)	HowMany(MotBk)	P	

HFCS 2012: Vehicles - Routing error log

Variable Name	Spec Page Ref	Scenario	Possible Error / Problem	Comment	To be resolved by
Cars_No	124	Cars_No = 0	Test plan says it should go to Oth_Veh but error message is generated saying range 1-99	Limits not stated in spec. Gerry said he intended to allow 0. Now says leave as is.	Denis
		Cars_No 1-4	Test plan says it should go to Oth_Veh but routes to Cars_Val.	Check Expected Result O.K. as per spec 6/11	Denis
Business	125 - 132	Business = 1 AND HowMany < 1	Test plan says HowMany < 1.	0 not allowed in coding but Gerry may have wanted it. Check with Gerry. Gerry said leave as is.	Denis
Other_V	133	Other_V = 0 AND Cars = 1	Test plan says route to Veh_Spnd but form routes to Veh_Buy.	Check Expected Result as Veh_Buy should be asked before Veh_Spnd. Error in TestPlan.	Denis
OthVehVal	133 - 136	Other_V = 2 AND Cars = 1	Test plan says route to Veh_Spnd but form routes to Veh_Buy.	Check Expected Result as Veh_Buy should be asked before Veh_Spnd. Error in TestPlan.	Denis
OthVehVal	133 - 136	Other_V = 1 etc. AND Cars = 2	Test plan says route to Jewels or Valuables but form routing to Veh_Buy	Recheck spec. and check Expected Result. O.K. as per spec 6/11	Denis
Filter		Cars = 1 AND Oth_Veh = 1	Test plan says ask Veh_Buy if Cars = 1 AND Oth_Veh = 1. Spec says as Veh_Buy if Cars = 1 OR Oth_Veh = 1.	Check Expected Result. Spec 6/11 states: Ask if Cars=1 or Oth_yeh=1	Denis

Test case design techniques for Blaise Code



- Systematic approach for developing test cases
- Generate test cases that have better chance of finding faults
- An objective method of developing test cases

Decision tables

Equivalence partitioning

Boundary Analysis

Use Case

flowcharts

State transition

Test case design techniques used for Routing test logs



- Decision tables proved a very useful tool for blaise testing
- Programmers encouraged to draw specifications in flow charts and state transition diagrams

Decision
tables

flowcharts

State
transition

Test case design techniques used for Ranges/computations test logs



- Mapping test cases using Equivalence partitioning helps to define representative values of valid and invalid ranges
- Boundary Analysis used to define and test minimum and maximum values of a range

Equivalence
partitioning

Boundary
Analysis

Test case design techniques used for Inserts & Question text Logs



- Use case or Scenario testing used for testing inserts and fills in Question text
- Incorporate into these tests were visual and Question text tests

Use Case

System & Integration testing



Who ?

- Developers



System & Integration testing



Types of tests:

- Black box testing
- Use Case scenario testing

System & Integration testing



Non functional requirements tested:

- Installability
- Maintainability
- Performance
- Load & Stress handling
- Recovery
- Usability

User Acceptance testing



Who ?

- Business Users
- Independent of Blaise and IT teams

User Acceptance testing



Types of tests:

- Use Case testing [scenarios]
- Pilot

Performance & Results



- Over 80 test log templates were prepared
- Test logs prioritized by complexity
- 3.5 independent testers took 15 -20 days to complete the logs
- Testing and re-testing continued until Questionnaire sign-off [1 week before release for pilot]

Performance & Results



- Testing documentation was reviewed and updated throughout development
- Extra testers if needed
- All incidents corrected, retested and signed off or waived

Results

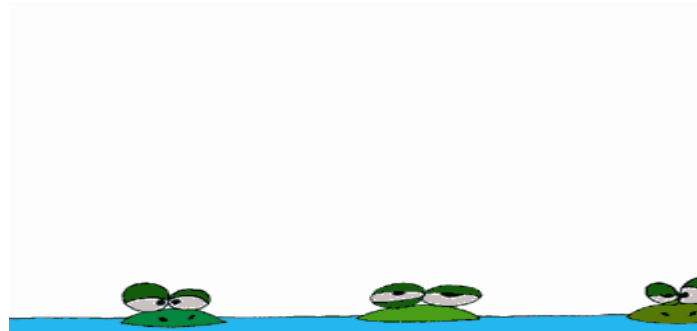


Results

- No critical problems in live environment
- Helpdesks calls related to the Questionnaire were Interviewer training issues
- Positive feedback from Business area on the Quality of the Questionnaire

Conclusion

- 25% of development time assigned to testing
- Creating and maintaining the large volume of test logs was time consuming but definitely worth the effort



Conclusion

- 25% of development time assigned to testing
- Creating and maintaining the large volume of test logs was time consuming but definitely worth the effort

Conclusion



Thank You