SysTest Labs
216 16th Street • Suite 700
Denver, Colorado 80202
tel: 303/575-6881 • fax: 303/575-6882
www.systest.com

DSDI Static Code Review and Analysis Findings Report

ASSESSMENT REPORT FOR
DRAEGER SAFETY DIAGNOSTICS, INC.
ALCOTEST 7110 MKIII-C.
NEW JERSEY FIRMWARE VERSION NJ3.11

Developed for:

NEW JERSEY SUPREME COURT

August 28, 2007

Table of Contents

Executive Summary

On July 1, 2007 SysTest Labs Incorporated, a Denver, Colorado based firm that specializes in software Quality Assurance and Testing, was engaged by Draeger Safety Diagnostics, Inc. (DSDI), to perform a static code review of the Alcotest NJ3.11 source code. Independent testing was to be performed pursuant to an Order of the New Jersey Supreme Court dated May 22, 2007 (The Order). This called for a complete examination of DSDI's Alcotest NJ3.11 source code, with particular emphasis placed on a review for "obvious issues within the code," and "consistency with the algorithms as documented in the software," (specified in Paragraph 1 of a letter of instruction between Samuel Sachs, Esq. and Jeffrey Schreiber, Esq. dated December 11, 2006). It is SysTest Labs' understanding that this letter has come to be referred to as "Addendum A," and that the terms of the letter are incorporated within The Order.

SysTest Labs Incorporated personnel reviewed software version NJ3.11 of the source code for the DSDI Alcotest 7110 MKIII-C. This code comprises the total software application of the Alcotest 7110 MKIII-C and is responsible for all results generated. We determined the following:

- The Alcotest NJ3.11 source code, as received by SysTest Labs from DSDI was the full and correct code base used on the Alcotest 7110 MKIII-C.

- The individual source code modules, or functions, as mapped by SysTest Labs' control flow diagrams, follow logical paths.

- Nowhere in the code did it appear that an attempt had been made to deliberately and/or maliciously alter or corrupt the software in order to generate inaccurate test results.

During our review, two (2) issues with the source code were discovered that, in our opinion, were outside the scope of The Order. Neither affects test results when the Alcotest 7110 MKIII-C system is used according to the manufacturer's guidelines, other than in one specific, unusual circumstance, as outlined below:

1. It was recognized that the examined source code is highly complex. Industry best practices dictate that cyclomatic complexity, a standard measure of source code complexity indicative of both understandability and maintainability; have a value no greater than 10. Upon review the NJ v.3.11 source code was found to include 81 modules with cyclomatic complexity indices in excess of 10 and three modules with indices in excess of 100.

2. A buffer overflow error was found that invalidates the "Reported Breath Test Result" of a subset of specific tests. These occur only under the following, well defined, conditions:

During tests in which the subject is required to provide three sufficient breath samples, and the third sample is within tolerance with each of the other two samples, and the lowest of the six recorded result values is the second breath sample's electrochemical test result, the “Reported Breath Test Result” is invalidated by a buffer overflow error, although the measured alcohol concentration values are correctly retained and reported in the Alcohol Influence Report.

It is the opinion of SysTest Labs that while the reviewed source code is not written in a manner consistent with usual software design best practices, there are no obvious defects intentionally written to produce anything other than consistent test results. SysTest Labs Incorporated expects that the Alcotest 7110 MKIII-C's source code, as written, and when used in accordance with DSDI guidelines, will reliably produce consistent test results.

Introduction

On July 1, 2007 SysTest Labs Incorporated, a Denver, Colorado based firm that specializes in software Quality Assurance and Testing, was engaged by Draeger Safety Diagnostics, Inc. (DSDI), to perform a static code review of the Alcotest NJ3.11 source code. Independent testing was to be performed pursuant to an Order of the New Jersey Supreme Court dated May 22, 2007 (The Order). This called for a complete examination of DSDI's Alcotest NJ3.11 source code, with particular emphasis placed on a review for "obvious issues within the code," and "consistency with the algorithms as documented in the software," (specified in Paragraph 1 of a letter of instruction between Samuel Sachs, Esq. and Jeffrey Schreiber, Esq. dated December 11, 2006). It is SysTest Labs' understanding that this letter has come to be referred to as "Addendum A," and that the terms of the letter are incorporated within The Order.

Methods

The code review and analyses requested by DSDI consisted of the following tasks and required the specified time to perform:

Task	Hours
1) Performed a preliminary examination of the source code gathering basic metrics such as the number, names and sizes of the included modules.	4 Hours
2) A visual, static, line by line code review was performed.	320 Hours
3) Control flow diagrams were produced using the Scientific Toolsworks, Inc. Understand C+ tool. Code tracing, reverse engineering, code navigation, and metrics applications were employed. (http://www.scitools.com/index.php)	120 Hours
4) A dynamic comprehensive code examination was performed to identify vulnerabilities within the code. The Fortify SCA code review tool was used. (http://www.fortifysoftware.com)	20 Hours
5) Source Code was compiled on 06/11/2007 according to instructions provided by Brian Shaffer of DSDI and BaseOne.	4 Hours
6) The check sum, or digital footprint of the source code, was compared to an ROM Image provided by the Sergeant Kevin Flanagan of the New Jersey State Police.	1 Hour
7) Hannsa One, a professional translation organization, was employed to translate German language components of the source code under review.	74 Hours
8) Research, compilation, and production of the final report.	56 Hours

Definitions

1. Source Code: Source code and object code refer to the "before" and "after" versions of a computer program that is compiled before it is ready to run in a computer. The source code consists of the programming statements that are created by a programmer with a text editor or a visual programming tool and then saved in a file.

2. Static Code Review: Static code analysis is a set of methods for analyzing software source code or object code in an effort to gain understanding of what the software does and establish certain correctness criteria.

3. Dynamic Code Analysis: Dynamic source code analysis is a technology technique aimed at locating and describing areas of weakness in source code through the use of an automation software tool. Those weaknesses might be security vulnerabilities, logic errors, implementation defects, buffer overflow errors, concurrency violations, or boundary conditions, which are problem-causing in the source code.

4. Buffer Overflow: A buffer overflow occurs when a program or process tries to store more data in a buffer (temporary data storage area) than it was intended to hold. Since buffers are created to contain a finite amount of data, the extra information - which has to go somewhere - can overflow into adjacent buffers, corrupting or overwriting the valid data held in them.

5. Cyclomatic complexity: Measures the amount of decision logic in a single software module. It is used for two related purposes in the structured testing methodology. First, it gives the number of recommended tests for software. Second, it is used during all phases of the software lifecycle, beginning with design, to keep software reliable, testable, and manageable. Cyclomatic complexity is based entirely on the structure of software's control flow graph.

6. Module Finder EX is a SysTest Labs developed tool. Its general purpose is to parse code in search of modules (i.e. classes and functions). Line counter is an added feature to Module Finder that reads selected files or folders and determines each individual files properties. The properties of the files are the number of source code lines, number of comment line, number of mixed code and comment lines, and the system size of the file as provided by Windows in KB.

References

1. Alcotest NJ3.11 source code
2. Addendum A
3. Alcotest 7110 MKIII-C New Jersey State Police User Manual-Technical
4. Source Code Build Instructions
5. National Institute of Standards and Technology
6. http://searchsecurity.techtarget.com/

Findings

The translation from German to English of the comments within the major components shows the logical intent of the programmers to produce reliable and valid test results. SysTest was unable to find any evidence of any intention to mis-direct or re-direct the test results or report anything other than valid results.

The Alcotest NJ3.11 source code appears to have evolved over numerous transitions and versioning, which is responsible for cyclomatic complexity.

SysTest's review reveals there is a Buffer Overflow issue in file ir_messw.c containing a condition which invalidates the "Reported Breath Test Result" of a limited subset of tests when conducted under defined conditions.

Appendix A

SysTest Labs

SysTest Labs Incorporated is a privately held, Colorado corporation, founded in 1996 and headquartered at 216 16th St., Suite 700, Denver, Colorado, 80202. We provide software Quality Assurance, Test Engineering, Security Analysis, Project Management, and Independent Verification and Validation (IV&V) services to private and public entities. Our clients range in size from small/ medium sized businesses to large state agencies and Fortune 100 corporations with global operations. We maintain a staff of over 100, comprised predominantly of test engineers and certified Project Management Professionals. On each of the over 450 projects we have completed we strived to meet the unique QA and Testing needs of our clients, wherever in the world they were required.

As an Information Technology (IT) services industry leader in software quality, SysTest Labs maintains a unique focus. In our on-site testing lab, we have developed rigorous methodologies that are embodied in our *Advanced Test Operations Methodology™ (ATOM™).* ATOM is based upon industry best practices and testing standards promoted by the Institute of Electrical and Electronics Engineers (IEEE), the Internal Organization for Standardization (ISO), and the Capability Maturity Model Integration (CMMI) standards of the Carnegie Mellon University Software Engineering Institute (SEI). In addition, SysTest adheres to the principles and guidelines of the Project Management Institute (PMI) as detailed in its publication *Project Management Body of Knowledge (PMBOK).*

Our methodologies have been audited and certified by independent, national agencies. Specifically, we are certified as one of only three National Voluntary Laboratory Accreditation Program (NVLAP) facilities. This required us to complete and pass rigorous audits by the National Institute of Standards and Technology (NIST). Our accreditation allows us to analyze, test, and certify electronic voting system software and hardware. States, counties and municipalities across the nation depend on us to certify that their electronic voting systems when manufactured are accurate, reliable, maintainable, usable and meet all statutory requirements and industry standards.

These principles form the core of our “industrial strength” approach and methodology to software testing and analysis. Additionally, we employ expert industry technologists and certified Project Management Professionals (PMP), as opposed to “generalist” consultants. Our staff is, first and foremost, comprised of software quality engineers who possess the specialized skills needed to test and analyze products across the full Software Development Lifecycle (SDLC). They are experts in the use of industry standard suites of tools such as IBM Rational®, and Borland® Software, among others.

Appendix B

Potential Code Issues:

The following describes these issues.

General Code Issues:

Not all conditional statements perform explicit comparisons. For example, many checks in the form of "if (x)" instead of "if(x==TRUE)". Explicit comparisons are generally required by good coding practices and not using them can introduce defects into code.

Non descriptive variable names – i, j, dummy and temp

Appendix C

Function Code Issues:

ir_messw.c
BANKED void berechne_aak_endergebnis(void)

Array aak is declared to the size of 4, which allows subscripts of 0 to 3. Lines 292 and 293 in the file ir_messw.c use array subscripts 4 and 5 for aak, which is forcing a write beyond the end of the declared array size. This is considered a buffer overflow, and does in certain circumstances result in the corruption of the "Reported Breath Test Result".

Global Variables:
Global variables are variables that are accessible from any function within a program. The benefit of global variables is that they can be used at any time without the need of passing them to a function so that they can be used. The negative of global variables is that the data they contain is not protected and that data can be changed at any time by any function in an application, intentionally or unintentionally. Because of this problem, the use of global variables should be extremely limited, and only used to non critical data in the application.

In the source being examined, test and result data is being stored in global variables. This data can be considered critical data, and should not be stored in this manner. It should be stored in variables or structures that are only passed to functions that require access to the data. This way the data is protected and can not be altered by other functions within the application.

Unused functions:
Functions that are not used and may be reserved for future or jurisdiction use

The following are functions that are never called in the application:

File Name	Function Name	Starting Line Number
Bedien.c	set_lockout_time	798
Datalog1.c	fillbuff_subrec4	051
Datalog1.c	GetSizeSubRec4	

File Name	Function Name	Starting Line Number
		092
Datalog1.c	StoreRecFlow2	
		632
Datalog1.c	StoreRecIr1	
		552
Datalog1.c	StoreRecVol3	
		712
Datalog2.c	have_space	
		27
Eing_dat.c	cadmv_to_strings	
		166
Eing_dat.c	GetMaleFemale	
		06
Eing_dat.c	LabTechToStrings	
		290
Eing_dat.c	PresLimit	
		342
Eing_dat.c	simtemp_on_off	
		17
Eing_dat.c	UserIdToStrings	
		208
Eing_dat.c	ValidYesNo	
		00
Global.c	global	
		98

File Name	Function Name	Starting Line Number
Heizung.c	atemsondeheizung_einschalten	43
Hpljet.c	Draw3LF_CR	54
Hpljet.c	DrawCheckedBox	44
Hpljet.c	set_aak_temp	796
Hpljet.c	set_temp	836
Hw_error.c	pruefe_kalwerte	57
Irq.c	irq_handler	50
Keyboard.c	magstrip_wait_string	15
Main7110.c	CheckRam	50
Maint.c	Maint	15
Memo_utl.c	ram_bank_next_on	80
Memo_utl.c	ram_check_all	24
Modemfkt.c	t_r_modem	

File Name	Function Name	Starting Line Number
Pctalk2.c	check_kbchar	35
Pctalk2.c	get_mp_nbytes	28
Pctalk2.c	init_sci_rcv_array	76
Prin_int.c	i_reset	46
Prin_int.c	printer_output	94
Protokol.c	Int2signedString	15
Protokol.c	print_aak_einheit	56
Start.c	start_prog	188
Uhr.c	DateTimeFormatStr	2
Uhr.c	DelayShowTime	93
Uhr.c	EpochDiffMinutes	83
Util.c	byte2str	061
		27

File Name	Function Name	Starting Line Number
Util.c	str2float	00
Util.c	str2word	41
Util.c	ubyte2str	02
Util.c	uword2str	60
Util.c	word2str	19
W_TIMING.S07	GERAET	
W_ZANFG.S07	wird	4
W_ZANFG1.S07	GERAET	
Xirq.c	i_xirq	67
Z_format.c	bcd2ascii	52
_eeprom.c	password_test	27
_eeprom.c	print_eeprom	39
_eeprom_T.c	password_test	24
_eeprom_T.c	print_eeprom	

File Name	Function Name	Starting Line Number
volume.c	aak_flow_correction	007
volume.c	volume_show	39
		21

Unused Objects:
Unused objects are variables that are may be reserved for future or jurisdiction use

File Name		Function Name	Location Line Number
Abgleich.c	antwort		124
Ablauf.c	answer		127
Ablauf.c	answer		831
Ablauf.c	equipData		834
Ablauf.c	j		127
Ablauf.c	minutes		127
Ablauf.c	textNum		124
Atem_tmp.c	text_1		95
Atem_tmp.c	text_2		96
Bedien.c	answer		1046
Bedien.c	answer		1126
Bedien.c	answer		1178
Bedien.c	anzahl		1046
Bedien.c	anzahl		1126
Bedien.c	anzahl		1178
Bedien.c	dest		1046

File Name	Function Name	Location Line Number
Bedien.c	dest	1126
Bedien.c	dest	1178
Bedien.c	solnCopyData	1128
Bedien.c	solnCopyData	1180
Bedien.c	value	1397
Datalog1.c	epoch	816
Datalog1.c	i	814
Datalog1.c	i	1081
Datalog1.c	i	1242
Datalog1.c	j	814
Datalog1.c	j	1081
Datalog1.c	j	1714
Datalog1.c	j	1798
Datalog1.c	j	1899
Datalog1.c	j	1975
Datalog1.c	j	2053
Datalog1.c	k	814
Datalog1.c	k	1242
Datalog1.c	k	1714
Datalog1.c	messung_nr	364
Datalog1.c	target	928
Datalog1.c	tmpStr	927
Datalog2.c	aacount	356
Datalog2.c	ptr	214
Ec_senso.c	ecCalLoadFactor	482
Eing_dat.c	answer	161

File Name	Function Name	Location Line Number
Eing_dat.c	answer	237
Eing_dat.c	arrstTimeString	328
Eing_dat.c	done	328
Eing_dat.c	epoch	331
Eing_dat.c	status	161
Eing_dat.c	status	422
Eing_dat.c	temp	162
Flow.c	flow_cur	559
Flow.c	flow_diff	559
Flow.c	flow_high_corr_val	559
Global.c	breathtemp_min	48
Global.c	cc_status	63
Global.c	dataAcq1	298
Global.c	dr_timing_alt	72
Global.c	ee_ad_kalfaktor	404
Global.c	ee_config_byte	409
Global.c	ee_ec_ref_integral_c	382
Global.c	ee_ec_ref_integral_d	383
Global.c	ee_not_used_1	386
Global.c	ee_not_used_2	422
Global.c	ee_not_used_3	461
Global.c	ee_not_used_4	462
Global.c	ee_not_used_7	412
Global.c	ee_nr_printouts_ext	476
Global.c	ee_unused_flag	477
Global.c	eeFreeBytes	415

File Name	Function Name	Location Line Number
Global.c	FreeByts	446
Global.c	FreeByts1	439
Global.c	irq_error	159
Global.c	keyb_buf1p	170
Global.c	keyb_buf2p	171
Global.c	keyb_key	173
Global.c	keyb_stat	172
Global.c	mouthpiece_temp	192
Global.c	rom_bank_no	216
Global.c	start_breathtemp	250
Global.c	test_count	260
Global.c	timing_impuls_counter	268
Hpljet.c	avgCntr	1293
Hpljet.c	batVolt	1290
Hpljet.c	breathTemp1	1290
Hpljet.c	breathTemp2	1290
Hpljet.c	CP_BOTTL	163
Hpljet.c	CP_COL1C	189
Hpljet.c	CP_COL2B	185
Hpljet.c	CP_COL3D	191
Hpljet.c	CP_COL_E	194
Hpljet.c	CP_HEAD2	155
Hpljet.c	CP_HEAD3	156
Hpljet.c	CP_INFO	168
Hpljet.c	CP_OP_NM	162
Hpljet.c	CP_USR2	196

File Name	Function Name	Location Line Number
Hpljet.c	epoch	1288
Hpljet.c	FONT_ARIAL	153
Hpljet.c	hose	1290
Hpljet.c	i	1117
Hpljet.c	i	1203
Hpljet.c	i	1285
Hpljet.c	jdFrac	1289
Hpljet.c	jdInt	1289
Hpljet.c	k	1285
Hpljet.c	kuev	1290
Hpljet.c	locPosition	1118
Hpljet.c	locPosition	1204
Hpljet.c	negAvg	1294
Hpljet.c	negPeak	1294
Hpljet.c	operMode	1285
Hpljet.c	p2pAvg	1294
Hpljet.c	passInit	1285
Hpljet.c	peak2Peak	1294
Hpljet.c	posAvg	1294
Hpljet.c	posPeak	1294
Hpljet.c	press	1291
Hpljet.c	probeTemp	1291
Hpljet.c	radioCuev	1286
Hpljet.c	radioDry	1287
Hpljet.c	radioDryCO2	1287
Hpljet.c	radioHose	1286

File Name	Function Name	Location Line Number
Hpljet.c	radioPort	1286
Hpljet.c	radioWet	1287
Hpljet.c	radioWetCO2	1287
Hpljet.c	rfi	1290
Hpljet.c	simTempBuff	1285
Hpljet.c	szDegCsym	305
Hpljet.c	szRadioChkd	303
Hpljet.c	szRadioOpen	302
Hpljet.c	szSpaceString	306
Hpljet.c	szSubjRfusChk	304
Hpljet.c	volt12	1290
Hpljet.c	voltsEc	1290
Ir_messw.c	i	209
Ir_prob.c	dummy_d_word	122
Ir_sampl.c	posHalf	139
Lin_test.c	k	239
Lin_test.c	text_nr_ec	240
Lin_test.c	text_nr_ir	240
Main7110.c	dummy	124
Main7110.c	i	124
Main7110.c	j	124
Maint.c	answer	291
Maint.c	i	291
Maint.c	k	291
Memo_utl.c	dummy	182
Mw_error.c	aak_differenz	439

File Name	Function Name	Location Line Number
Mw_error.c	aak_median	439
Mw_error.c	aak_percent10	439
Pctalk1.c	i	119
Prin_int.c	block	518
Prin_int.c	zeichen	717
Protokol.c	ptr	417
Read_dat.c	j	755
Read_dat.c	j	835
STDLIB.H	quot	26
STDLIB.H	quot	32
STDLIB.H	rem	27
STDLIB.H	rem	33
Service.c	avgNoise	614
Service.c	i	105
Service.c	j	105
Soln_cng.c	answer	148
Soln_cng.c	answer	472
Soln_cng.c	loc_simtemp	471
Soln_cng.c	manual	148
Soln_cng.c	manual	472
Soln_cng.c	no_after_pnt	431
Soln_cng.c	no_of_digits	431
Soln_cng.c	solnCngJdFrac	125
Soln_cng.c	solnCngJdInt	125
Soln_cng.c	text_nr_ec	150
Soln_cng.c	text_nr_ec	474

File Name	Function Name	Location Line Number
Soln_cng.c	text_nr_ir	150
Soln_cng.c	text_nr_ir	474
Uhr.c	dow	561
Uhr.c	merFlg	561
Uhr.c	promptText2	1205
Uhr.c	time	561
Util.c	vorzeichen	204
Util.c	vorzeichen	262
_eeprom.c	i	542
_eeprom.c	ok	411
_eeprom_T.c	i	1012
_eeprom_T.c	ok	764

Unused Types:
Unused Types are structures that are declared but may be reserved for future or jurisdiction use.

File Name	Function Name	Location Line Number
STDARG.H	va_list	37

Cyclomatic Complexity:

Cyclomatic Complexity is a software measurement that was developed by Thomas McCabe and is used to measure the complexity of application code. It measures the number of independent paths through an application's code.

To fully test a function within an application, a test case would be needed that would fully execute every path through the function. This means that a function with a high complexity will require a larger number of tests to fully test the function. It also means that functions with a high complexity are more difficult for a programmer to understand since they must understand every pathway and the results produced by executing each path.

With higher complexity comes a higher risk to the code for inherent defects, as well as a higher risk for bad defect fixing, which is to say that the fix for a found defect may not actually fix the defect, could introduce new defects to the code, or both.

Coding standards guidelines recommend keeping the cyclomatic complexity of functions under 10, and many recommend keeping it under 7. There are exceptions to this rule, however. Some algorithms are inherently complex and do require a lot of decision points within them, so it is possible to have an algorithm that is simple to understand, but has a high complexity.

Some causes of a high cyclomatic complexity are:

1. Functions that perform more than one action.
2. Functions handling states resting in global variables instead of a good state machine.
3. Intentional or unintentional use of brute force code instead of algorithms.
4. Lack of design patterns.

The best overall solution for lowering cyclomatic complexity is to examine functions with a high complexity and refactor the code to make it less complex. This will generally involve breaking the code into multiple smaller functions based around actions the code performs. This increases the size of the function call tree, but will make the code much easier to understand and maintain.

The following is a list of all the functions in the application and their cyclomatic complexity, ordered from most complex to least:

Function	Cyclomatic	Nesting
make_mess_block	105	5
send_record_descriptors	103	7
messablauf	102	10
bedienung	82	5
main	61	9
ir_probenahme	60	5
start_communicating	56	9
volume_show	53	4
ReceiveOperatorUpdate	51	8
store_rec_meas_1	51	3
receive_new_program	50	8
make_printer_block	49	7
i_xirq	46	5
alkohol_messung	40	6
HpljTestSeq	39	3
single_test	37	6
ir_ec_abgleich	34	4
air_blank	32	6

Function	Cyclomatic	Nesting
pruefe_differenz	31	2
heizungsregelung	27	5
LinTest	26	3
SolutionChange	26	3
control_flow_into_cuvette	24	4
Maint	23	5
ec_calculate	22	4
RecMeas1mem2ds	22	1
send_all_data	22	6
flow_abgleich	21	5
atemtemp_abgleich	20	4
CalGasUeberpruefung	20	4
ControlTest	20	3
labor_protokoll	20	3
pruefe_interferenz	20	3
TimeString2Epoch	20	3
druck_abgleich	19	3
Epoch2TimeString	19	4
ir_periodenwerte	19	5
one_subj_test	19	3
berechne_aak_endergebnis	18	3
berechne_ir_aak	18	4
LookupByTstNum	18	4
store_rec_lin_1	17	3
ControlFlowViaB_Hose	16	4
gas_einstellen	16	3
GetDateTimeString	16	7
SetClock	16	4
word2str	16	5
control_flow_into_port	15	4
copy	15	5
store_rcv_bytes_uart0	15	5
store_rcv_bytes_uart1	15	5
storeRecCtrlSoln1	15	3
byte2str	14	4
eingabe_daten_proband	14	5
float2str	14	4
resultat_formatieren	14	3
druck_flow_abgleich	13	3
ShowIrNoise	13	5
to_ram	13	7
check_received_string	12	6
CtrlSolnCopy	12	4

Function	Cyclomatic	Nesting
eingabe_daten_arrest	12	7
GetDateString	12	6
LinCopy	12	4
SimulatorToTemp	12	5
druck_flow_anzeige	11	3
eingabe_daten_general	11	7
eingabe_daten_rems	11	5
GetTimeString	11	4
HpljCalibCtrlSoln	11	3
Int2signedString	11	2
keyb_wait_string	11	3
make_value_string	11	2
read_password	11	3
set_text	11	3
show_ir_periodenwert	11	4
auto_set_probe	10	3
DateString2Epoch	10	3
DaylightSavings	10	1
ec_referenz_speichern	10	3
flow_mess	10	3
ir_monitoring	10	2
IrCalGasMessung	10	3
service_uart0	10	2
service_uart1	10	2
t_r_modem	10	3
dial_out	9	2
do_monitor_simtemp	9	4
get_nbanks	9	1
ir_detektor_drift_pruefung	9	3
PrintBreathTest	9	4
PrintCalSolInt	9	3
PrintLinTest	9	4
set_printer	9	3
setString	9	3
str2float	9	2
check_breathprobe_temp	8	3
GetAbsTolerance	8	3
GetCalGasInlet	8	3
GetTargBacConc	8	3
init_modem	8	2
ir_nullwert_messung	8	3
irq_handler	8	2
ndigits_toggle	8	3

Function	Cyclomatic	Nesting
pressure_flow	8	2
PRINT	8	3
PRUEFE_NTC_HEIZUNG	8	2
str2word	8	2
ubyte2str	8	3
unit_einstellen	8	4
uword2str	8	3
volume_integral_get	8	3
AbaGasSelection	7	3
add_password	7	5
add_password	7	5
ec_motor_nullpunkt	7	4
Epoch2DateString	7	3
GoodSetClkEpoch	7	2
magstrip_wait_string	7	3
PrintCalTests	7	3
pruefe_umgebungsluft	7	3
PruefeCalGas	7	2
resultat_runden	7	3
sensor_einlesen	7	3
umgebungsluft_pruefung	7	3
UserIdToStrings	7	1
ValidConcenString	7	4
breathtemp_calculate	6	3
cadmv_to_strings	6	1
check_file_crc	6	4
check_kbstring	6	3
CheckRam	6	3
ChooseDateFormat	6	3
ChooseHoseType	6	3
ChooseTimeFormat	6	3
flow_anzeige	6	4
HoseSet	6	1
keyboard_eingabe_vergleich	6	2
LinePrinter_Data	6	2
make_more_text	6	2
messwert_im_aak_feld_ablegen	6	2
mw_error_handler	6	2
printer_output	6	5
read_number	6	4
SecurityKey	6	3
SecurityKey	6	3
send_rec_head_1	6	1

Function	Cyclomatic	Nesting
show_value_from_adc	6	3
ShowCombinedFlow	6	4
ShowIrEnergy	6	3
sim_power_on_off	6	3
simtemp_on_off	6	3
store_rec_brth_1	6	1
store_rec_stnd_1	6	1
untermenue	6	2
Adjust12V	5	3
berechne_ec_sensor_aak	5	2
check_kbchar	5	3
check_mem_ok	5	2
copy_string	5	1
CounterEinlesen	5	3
del_password	5	3
del_password	5	3
DelayShowTime	5	4
display_write_string	5	3
eing_dat_on_off	5	3
eingabe_daten_user	5	5
error_signal_1	5	2
get_key_start	5	4
get_rec_type	5	2
get_text_input_nr	5	1
GetDynamicData	5	1
HoseTemp	5	1
HpljLinTest	5	3
init_uart_tx	5	2
ir_perioden_mittelung	5	2
Jd2Epoch	5	1
kalwert_einlesen	5	3
labor_protokoll_ein_aus	5	3
move_ptr	5	2
next_rec	5	3
password_syntax	5	2
password_syntax	5	2
PresLimit	5	3
printer_paper	5	4
QuadraticAdjust	5	3
read_no_tests	5	3
read_string_from_keyboard	5	2
read_y_s_char	5	3
send_x_packet	5	2

Function	Cyclomatic	Nesting
SendStatic	5	1
set_probe	5	3
SetAccGasConcTol	5	2
SetCal1GasConcTol	5	2
SetEcGas	5	3
SetSimEqulibTime	5	3
SetSolnCngDaysLimit	5	3
SetSolnCngTestLimit	5	3
store_rec_meas	5	2
StoreRecVol3	5	2
str2uword	5	2
TwoChoice	5	2
wird	5	0
aak_flow_correction	4	3
adjust_rfi_limit	4	3
AmbigousEpoch	4	1
breathtemp_sample	4	2
check_warn	4	2
eingabe_anzahl_ausdrucke	4	3
fillbuff_subrec3	4	1
flow_ueberwachung_null	4	2
FromRam	4	3
GERAET	4	0
GetFlow	4	2
GetSimtemp	4	2
GetSubjectAge	4	2
GetTexteStrLen	4	2
HW_ERROR_HANDLER	4	1
LabTechToStrings	4	1
left_just_string	4	2
location_einlesen	4	2
message	4	3
password_test	4	3
password_test	4	3
pot	4	2
PrintSimTempInt	4	3
pruefe_druck_temp	4	1
PRUEFE_DRUCKSENSOR	4	2
PRUEFE_FLOW_SENSOR_1	4	2
PRUEFE_HEIZUNGS_REGELUNG	4	2
PRUEFE_U_BATTERIE	4	2
PRUEFE_U_VERSORGUNG	4	2
ram_check_all	4	1

Function	Cyclomatic	Nesting
seriennummer_einlesen	4	2
set_aak_temp	4	2
set_modem_number	4	2
SetCal2GasConcTol	4	2
SetRelTolerance	4	3
show_mem	4	2
ShowIrPosNeg	4	3
ShowIrSignal	4	3
store_rec_msg_1	4	1
trm_next_byte	4	2
aak_feld_marke	3	2
atemtemp_anzeige	3	2
bin2ascii	3	2
DateTimeFormatStr	3	1
druck_anzeige	3	2
Epoch2Jd	3	1
fillbuff_subrec1	3	1
funktionsstecker_farbe	3	2
funktionsstecker_pruefung	3	1
GERAET	3	0
GetMaleFemale	3	1
GetTexteStrPtr	3	2
GetYesNo	3	1
GoodDate	3	1
have_space	3	2
HpljPrintQc	3	1
HpljPrintTest	3	1
init_prob_protokoll_variablen	3	1
init_protokoll_variablen	3	1
init_uart_rx	3	1
IsSolnCngTime	3	1
local_print	3	2
modem_dial_out	3	2
modem2code	3	2
print_aak_value	3	1
print_passwords	3	2
print_passwords	3	2
print_protokoll	3	2
print_text_und_messwert	3	1
PrintIntSolnData	3	1
PrintPageTwoHeader	3	2
RecSoln1mem2ds	3	1
scroll	3	2

Function	Cyclomatic	Nesting
set_error_protokoll	3	2
set_error_protokoll_attempts	3	2
set_temp	3	2
show_ad_wandler_wert	3	2
show_ntc_atemsonde	3	2
show_ntc_kuevette	3	2
show_ntc_schlauch	3	2
store_rec_brth	3	2
Str2Conc	3	1
toggle_signal_1	3	2
ValidHeight	3	2
ValidNumberString	3	2
wait_for_ec_sensor_ready	3	2
aak_breathtemp_correction	2	1
atemtemp	2	1
blank_dotline	2	1
breathtemp_aprox	2	1
CdrvCrc16	2	1
check_ec_offset	2	1
check_eeprom_pruefsumme	2	1
communicate	2	1
data_to_dotline	2	1
display_clear	2	1
display_modem_number	2	1
display_set_cursor	2	1
display_write_char	2	1
dsp_ready	2	1
ec_offset_messen	2	1
ec_referenz_korrigieren	2	1
EcAlcSumIncrease	2	1
ENDLOS_SCHLEIFE	2	1
EpochDiffMinutes	2	1
erase_mem_subject	2	1
erase_mem_user	2	1
flow_zeroline	2	1
flprog_to_common_ram	2	1
get_mp_code	2	1
get_mp_nbytes	2	1
get_pressure_flow_zero	2	1
GetSavingsStatus	2	1
init_breathtemp_measure	2	1
init_modem2	2	1
init_parameter	2	1

Function	Cyclomatic	Nesting
init_passwords	2	1
init_passwords	2	1
init_sci_rcv_array	2	1
init_sci_trm_array	2	1
InitVariables	2	1
make_einheit	2	1
make_line	2	1
make_pruefsumme	2	1
print_name_und_nr	2	1
print_units_and_time	2	1
PrintUserDataInt	2	1
ProbeTemp	2	1
protokoll	2	1
pruefe_ir_nullwert	2	1
pruefe_ir_perioden_mittelwert	2	1
pruefe_kalwerte	2	1
pruefe_umgebungsdruck	2	1
pruefe_umgebungstemp	2	1
ram_bank_next_on	2	1
RecLin1mem2ds	2	1
Rtc2Epoch	2	1
RtcReadCounterRegister	2	1
RtcWriteCounterRegister	2	1
send_dispchar	2	1
send_dispstring	2	1
send_printstring	2	1
send_setcurs	2	1
SendDynamic	2	1
set_aak	2	1
set_or_check_eeprom_pruefsumme	2	1
set_or_check_eeprom_pruefsumme	2	1
set_simtemp_protokoll	2	1
set_string	2	1
short_delay	2	1
ShowSimtemp	2	1
ShowSimtemp2	2	1
single_cal_test	2	1
single_subj_test	2	1
store_rec_calib	2	1
store_rec_ctrl	2	1
store_rec_lin	2	1
store_rec_msg	2	1
store_rec_soln	2	1

Function	Cyclomatic	Nesting
store_rec_stnd	2	1
SubRec1mem2ds	2	1
SubRec2mem2ds	2	1
tablecrc	2	1
TexteStrCpy	2	1
valid_reply_id	2	1
ValidMaleFemale	2	1
ValidYesNo	2	1
verify_crc1	2	1
verify_crc2	2	1
wait_for_i_xirq	2	1
wurzel	2	1
aak_korrektur_kuevetten_temp	1	0
aak_messwert_anzeige	1	0
aak_two_displayed	1	0
ascii2bcd	1	0
atemsonde_kalwert_einlesen	1	0
atemsondeheizung_ausschalten	1	0
atemsondeheizung_einschalten	1	0
AtemsondeSet	1	0
atemtemp_1	1	0
atemtemp_2	1	0
atemtemp_kalwerte_einlesen	1	0
bcd2ascii	1	0
bcd2bin	1	0
beep	1	0
bin2bcd	1	0
calc_crc	1	0
clear_mem	1	0
clr_printer_needle	1	0
control_mode_einstellen	1	0
CRC16	1	0
CuvetteSet	1	0
disable_modem	1	0
disable_pcdirect	1	0
disable_uart_hardware_interrupts	1	0
display_serialno	1	0
display_version	1	0
display_write_set_string	1	0
DisplayDateTime	1	0
Draw3LF_CR	1	0
DrawCheckedBox	1	0
DrawLF_CR	1	0

Function	Cyclomatic	Nesting
druck	1	0
druck_flow_kalwert_einlesen	1	0
druck_kalwerte_einlesen	1	0
druck_temp_korr_abgleich	1	0
druck_temp_korr_messung	1	0
dsp_control	1	0
dsp_control_init	1	0
ec_kalwert_einlesen	1	0
EcAlcSumReduce	1	0
fillbuff_subrec2	1	0
fillbuff_subrec4	1	0
fkt_nicht_moeglich	1	0
flow_kalwert_einlesen	1	0
ForceSolnChange	1	0
funktion_ausgefuehrt	1	0
get_breathtemp	1	0
GetEquipData	1	0
GetSizeSubRec1	1	0
GetSizeSubRec2	1	0
GetSizeSubRec3	1	0
GetSizeSubRec4	1	0
GetSolnData	1	0
global	1	0
i_reset	1	0
init_display	1	0
init_printer_output	1	0
init_ram_check	1	0
init_to_receive	1	0
init_uart	1	0
init_xirq	1	0
InitCounters	1	0
InitEquipData	1	0
ir_kalwert_einlesen	1	0
keine_einstellung	1	0
kuevettentemp_kalwert_einlesen	1	0
kurz_protokoll	1	0
KuveTemp	1	0
LinePrinter_Init	1	0
location_anzeigen	1	0
lowpass	1	0
make_ergebnisrahmen	1	0
make_messwert_zeile	1	0
make_print_line	1	0

Function	Cyclomatic	Nesting
modem_hangup	1	0
modem_to_command_mode	1	0
NextSolnCngTime	1	0
normal_mode_einstellen	1	0
papier_vorschub	1	0
print_aak_einheit	1	0
print_datum_und_uhrzeit	1	0
print_eeprom	1	0
print_eeprom	1	0
print_pointer	1	0
print_text_messwert_text	1	0
PrintHeaderInt	1	0
PrintInstData	1	0
PrintInstDataEx	1	0
PrintPageNum	1	0
PrintSolnData	1	0
PrintUserData	1	0
pruefe_hardware	1	0
ram_check	1	0
ready_signal_1	1	0
RecCalibMem2ds	1	0
RecCtrlMem2ds	1	0
RecLinMem2ds	1	0
RecMeasMem2ds	1	0
RecSolnMem2ds	1	0
reset_mem_pointers	1	0
rjust_string	1	0
schlauchheizung_ausschalten	1	0
schlauchtemp_kalwert_einlesen	1	0
sel_ram_bank	1	0
send_byte	1	0
send_byte_mp	1	0
send_rec_head	1	0
send_string	1	0
set_cal_no_tests	1	0
set_eeprom_pruefsumme	1	0
set_eeprom_pruefsumme	1	0
set_lockout_time	1	0
set_printer_needle	1	0
set_ram_bank	1	0
set_subj_no_tests	1	0
SetAbaGasType	1	0
SetAccGasInletGasType	1	0

Function	Cyclomatic	Nesting
SetCal1GasInletGasType	1	0
SetCal2GasInletGasType	1	0
show_adc_rfi	1	0
show_ec_offset	1	0
simtemp_kalvalue_set	1	0
start_prog	1	0
StoreEquipData	1	0
StoreRecFlow2	1	0
StoreRecIr1	1	0
StoreSolnData	1	0
user_signoff	1	0
user_signoff	1	0
user_signon	1	0
user_signon	1	0

Appendix D

Execution Roadmap:

The core of the application's functionality is encapsulated in a few large functions. The logic of those functions is described in pseudo code, descriptive English narrative indicative of the logical flow implemented within the functions.

The application's main driving function, the function that is triggered to run by default at system start-up, and runs continually until the system is shut down is, as required by the constraints of the programming language, named "main." Function main contains the logical framework of the application, and is described in attachment "Draeger - Function main described.txt." In function main the core of the application's logic can be seen to include initialization, configuration, testing and termination processes.

Function main includes a call to function "bedienung", which encapsulates the system's user interface functionality. Function bedienung is written as a series of short, conditional code segments each of which causes execution to branch to specific subfunctions of the application. The logic of function bedienung is described in attachment "Draeger – Function bedienung described.txt."

The logic of the execution of a breath test is encapsulated in function "messablauf" (measure expiration), which is called by function main. Function messablauf includes the code to cause initialization of the system's test related data objects, measurement of the breaths, storage of the test results, and the results report to be printed. The logic contained in function messablauf is described in attachment "Draeger - Function messablauf described.txt."

Function messablauf includes a call to function "alkohol_messung" (alcohol measurement) which contains the logic related to measuring the breaths' alcohol content. Function alkohol_messung contains the code for initializing the system for a subject's test, purging the cuvette, sampling the ambient air, capturing the IR and EC values, and calling for results to be calculated.

Function alkohol_messung directly calls functions "berechne_ir_aak" (calculate IR alcohol concentration) and "berechne_aak_endergebnis" (calculate final alcohol concentration). Function alkohol_messung also indirectly, through function "ec_calculate", calls function "berechne_ec_sensor_aak" (calculate EC sensor alcohol concentration). The three functions, berechne_ir_aak, berechne_ec_sensor_aak and berechne_aak_endergebnis contain the core of the logic related to computing alcohol measurement test results, and are described in attachments:
"Draeger - Function berechne_ir_aak described.txt", "Draeger - Function berechne_ec_sensor_aak described.txt" and "Draeger - Function

berechne_aak_endergebnis described.txt."

It is the opinion of SysTest Labs, and in concurrence with software industry standards, that these Execution Roadmaps should be held as proprietary to DSDI and therefore will be made available to the Special Master *in camera* for his review to the extent the Special Master deems it necessary or appropriate in accordance with the confidentiality agreement the Special Master approved.

Appendix E

Logical Diagram:

The control flow diagrams, as generated by the source code static analysis application, Understand for C++, are schematic representations of the logical flows within each individual function of the application. They are illustrative of the complexity of the overall application, and of each function in specific.

However, it is the opinion of SysTest Labs, and in concurrence with software industry standards, that these control flow diagrams should be held as proprietary to DSDI and therefore will be made available to the Special Master *in camera* for his review to the extent the Special Master deems it necessary or appropriate in accordance with the confidentiality agreement the Special Master approved.

Appendix F

Resumes

Brian Phillips

Title: President & CEO

SUMMARY OF SKILLS AND QUALIFICATIONS

- Over 26 years of software test engineering experience.
- Responsible for founding and running a 105 person outsourced, software test lab.
- Developed and published numerous software test methodologies for client-server testing, standalone application testing, large scale telecommunication system testing, and e-business/e-commerce testing.
- Expert in automated testing using several different vendor products, automated test management using several different vendor products, and requirements management, analysis and traceability.

ACCOMPLISHMENTS –SYSTEST LABS' PROJECT EXPERIENCE

(The following is a subset of the project experience over the last 7 years).

- Manager responsible for IV&V, system testing, configuration management and certification of a network-based call center application, based on an Edify implementation that allowed communication between a Fortune 50 telecommunication client's in-network systems and out-of-network customer resources. Additional responsibilities required management, coordination, and certification of this system on the Client's world wide in-service network.
- Manager responsible for system test and in-service network certification support of a voice over Internet call center/customer service access system.
- Manager responsible for IV&V, project and test management of a large, complex call routing/call management/call parking system, using Genesys, for a Fortune 50 telecommunication client. Responsibilities included coordinating in-service network certification and official release to production.
- Manager responsible for System testing and test management of MCI's full service network-based call center system (NGSN). MCI's solution was based on systems provided by Amerex (now Comverse Network Systems, Voice Services Platforms Division).
- Support manager for IV&V of Colorado's Unemployment Insurance IVR and Call Center implementation of Genesys and Siebel.
- Testing manager for automated functional and performance testing of a large, complex Siebel based internet and call center system for one of the largest television in-home shopping networks.
- Responsible for management, test planning, test execution, defect reporting and tracking for a Java based client-server application. This was a 4 person, 6 month project. Supported release of this application into general availability.

- Responsible for management, test planning, test execution, defect reporting and tracking for a C++ based client-server application. This was a 6 person, 2 month project. Supported release of this application into general availability.
- Responsible for management, test planning, test execution, defect reporting and tracking for a Windows 95 based, standalone application that was converted from a Macintosh environment. This was a 5 person, 3 month project. Supported release of this application into general availability.
- Responsible for management, test planning, test execution, defect reporting and
- Responsible for management and initial load test planning on an internet based e-business application. The application was an on-line shopping system. This was a 4 person, 3 month project. Supported release of this application into production use.
- Responsible for management, test planning, test execution, defect reporting and tracking for an internet based, on-line grocery order system. This was a series of 2, 2 person, 1 month projects. Supported release of these application into production use.
- Responsible for management, test planning support, test execution support, defect reporting and tracking for a large, complex internet portal system. This was a 5 person, 1 month project. Supported release of this application into production use.
- Responsible for management, test planning, test execution, defect reporting and tracking for a conference call center system. This was a 3 person, 2 month project. Supported release of this application into production use.
- Responsible for management, test planning, test execution, defect reporting and tracking for a tool set designed to facilitate the development of e-business applications. This was a 5 person, 2 month project. Supported release of this application into the next customer phase of integration.
- Responsible for management and initial test planning for a set of complex e-commerce bundled applications. This was a 7 person, 3 month project. Supported release of this application into production use.
- Responsible for management, design and execution of all manual test plans, automated tests, foreign language tests, and load/performance tests for a CD read/write, Windows Explorer based application. This was a series of 5, 6 person, 2 month projects.
- As Director of Test Operations for all of SysTest Labs' projects, responsibilities included overall management and review of all test processes, test architecture designs, and documentation.

- Consultant
  May 1995 – June 1996:
  Software quality assurance consultant contracted to U S West MIS. Responsibilities included all software testing, manual and automated, on a complex, Windows and Macintosh based client server system. In addition, responsible for establishing software test processes, detailed quality assurance techniques, and software development processes.

- Consultant
  November 1993 - May 1995:
  As a software quality assurance consultant contracted to U S West Communications, worked as the Software Test Lead responsible for software test planning, engineering, and execution on Unix based, client server CAD/Expert Draft and Customer Service Point of Contact

applications. Participated and led efforts to re-engineer U S West test methodologies, detailed quality assurance techniques, and software test planning activities. Responsible for a four member software test team.

Responsible for organizing and structuring software test teams, providing these teams with the appropriate test processes, and supporting their initial test efforts.

- Consultant
  August 1993 - November 1993:
  As an independent consultant, responsible for software test engineering, business development, marketing strategies, business presentations, business growth, and development, research, and documentation of proprietary test methodologies used in consulting.

- President and Chief Consultant
  July 1991 - July 1993: Quality Software Engineering, Inc.
  QSE was an information systems consulting firm dedicated to providing expertise in software engineering and software testing. As founder, co-owner, and President, responsible for business development and management, marketing strategies, business presentations, business growth, and development, research, and documentation of proprietary test methodologies used in consulting.

  Provided software test management consulting to Security Life of Denver. Evaluated entire software test department and systems development department and proposed recommendations on how best to improve the quality of on-going software test efforts. Personally developed and presented a software test methodology uniquely suited for the test and development environment at Security Life of Denver.

  Manager of an independent contract software test team at Storage Technology Corporation. Responsible for development of the Software Master Test Plan, Requirements Traceability and Validation Matrix, functional Verification/Test Requirements, Test Cases and Procedures, and complete management of the independent test methodology for testing the software for a RAID peripheral's maintenance control processor. Maintained the Product Functional Specification for the development team, enforced software engineering practices amongst the development team, consulted StorageTek management in improving the current development methodology, implemented a software error reporting process and system, and supported the development of a hardware emulator for use in testing.

- Consultant
  November 1990 - July 1991:
  Responsible for the development of automated test scripts (using a U S West proprietary scripting language) to be executed on an Automated Test Facility. Responsible for developing Software Requirements Verification/Traceability Matrix for the application, performing specification and design analysis, developing new test procedures, developing regression test suites consisting of manual and automated tests, and the execution and close-out of the application product tests.

- Lead Systems Analyst, In-house Consultant
  January 1990 - November 1990: Martin Marietta Astronautics Group (MMAG), Denver, Colorado
  Defined MMAG-wide requirements for a pilot Product Configuration Management Information System (an Oracle relational database). Defined requirements and the conceptual database design for an Engineering Design and Parts List database (an Oracle

relational database). Supported development of the five year long range plan for MMAG Business Management Information Systems.

- Lead Systems Test Engineer
  June 1981 - November 1985 and February 1987 December 1990: MMAG, Denver, Colorado
  Analyzed, verified, and validated automatic control system requirements, design specifications, and flight software. Analyzed, verified, and validated redundancy management (fault tolerant) software functions. Defined the requirements and managed the design and development of a six degree-of-freedom spacecraft simulation. Managed and developed controls software test schedules, software analysis tools, and software test plans and procedures. Responsible for seven Systems Test Engineers/Analysts and Controls Engineers. Assisted in the development and refinement of independent test methodologies.

- Adjunct Faculty Member
  April 1989 - 1996: University of Denver, University College, Denver, Colorado
  Requested by the University of Denver, University College to develop and teach a graduate level course in Software Systems Testing. Required extensive independent research during free time hours to ensure development of a class that was consistent with current industry standards, challenging for the students on a graduate level, and covered all pertinent and practical information.

- Senior Systems Integration Consultant
  November 1985 - February 1987: Martin Marietta Information and Communication Systems, Washington, D. C.
  Developed the three year hardware system implementation plan for FAA Flight Service Automation Systems (FSAS) throughout the continental United States. Conducted on-site hardware and software Acceptance Tests. Supported the development of the FSAS Prime Contractor's Statement of Work for the FAA.

PLATFORMS, ENVIORNMENTS, and TOOLS

Platforms
Unix Workstation, Windows Based PCs, Macintosh PCs, Tandem, Elxsi Mini, IBM and CDC Mainframes
Environments
Internet Based Applications, Client Server, PC, Embedded Systems, Mainframe, Oracle, OSF/Motif, C, C++, Cobol, Visual Basic, Java
Operating Systems
HP-UX, Sun Solaris, Tandem Gardian, Other Unix, Linux, all Windows, Macintosh OS, OS/2, TSO MVS
Tools
IBM Rational's Test Studio and RequisitePro, Segue's SilkTest and SilkPerformer, Automated Test Facility (OS/2 based), Mercury's WinRunner, LoadRunner and X-Runner, Compuware's QA Run and QA Load, RadView's WebLoad

EDUCATION

- Certified Software Quality Engineer, ASQ
- Masters of Computer Information Systems, University of Denver, University College, Denver, CO, March 1989.
- Graduate Courses in Electrical Engineering, University of Colorado at Denver, Denver, CO, 1984.
- Bachelors of Science, Aerospace and Ocean Engineering, Virginia Tech (VPI&SU), June 1981.

Geoffrey Pollich

Title: Senior Project Director

As SysTest Labs Director of Automation, Mr. Pollich's duties have included day-to-day project management on projects that have varied in scope from DOD and EAC compliancy to decision making and logical algorithm quality assurance assignments. He has been responsible for driving the Standard Lab Procedures in the areas of delivery, Voting, IV&V, and a reviewing the SysTest Labs ATOM methodology.

Mr. Pollich served as the Director of Quality Assurance at Gartner, Inc. in Stamford, CT and as the Acting Quality Assurance Director at Inovant/Visa USA in Foster City, CA. previous to accepting his present position as Senior Project Manager at SysTest Labs in Denver, Colorado. After leaving Gartner, Inc. Mr. Pollich establish the very successful GRP Group consulting firm where his title and duties were those of a Quality Architect, Mr. Pollich and his associates analyzed and implemented Software Development Life Cycle best practices within numerous Fortune 500 companies. Mr. Pollich is a methodology expert who has consulted and presented before Executive Boards such as WebMD, Jeppesen, Razorfish, Inc., Gartner, Inc. and Visa USA, regarding cutting edge methodological processes used in today's development environments. He has been a key presenter at the Software Development Quality Assurance Association of Denver on methodology and automation best practices.

As a SixSigma trained manager, and in his role as Gartner, Inc.'s Quality Assurance Director, Mr. Pollich has been a vendor solution decision maker on various enterprise level projects. He has often been tasked with proof of concept through implementation and compliance of vendors on his many mission crucial projects assignments with Fortune 500 companies.

Technical Experience

Applications:
MS Project; PowerPoint; Excel; Minitab; Visio; WORD; FrontPage; Microsoft Office
Software/Languages:
Applied knowledge of HTML, Perl, Java, C++, Visual Basic
Databases:
Oracle R11.5.9i; SyBase; Informix; SQL Server
Management, Development and Test Tools:
Applied knowledge of Mercury Interactive Test Director, WinRunner, QuickTestPro, Loadrunner; Segue QA Partner for workstation & server testing, Segue Silk for internet application testing; Applied knowledge of IBM/Rational TestManager, Rational Robot, RequisitePro, ClearQuest, ClearCase, Robot
Development and Test Methodologies:
Waterfall methodology, Rational Unified Process, Agile;
Rapid Application Development methodology, Extreme Programming
Operating Systems:
DOS, Windows, MAC/OS10, HP Unix, Sun/Solaris

Industry Experience:
Support Manager; Testing Analyst; Quality Assurance Manager; Senior Analyst; Director of Quality Assurance; Quality Assurance Consultant; Quality Architect

Project Experience

SysTest Labs

11/2005 – Present

Cuyahoga County, Ohio, Team Lead
As team lead on a compliance project for Cuyahoga County, Ohio in regards to their the Diebold voting equipment Mr. Pollich we responsible for reviewing processes of the Ohio Board of Elections, He interviewed all Election Board directors and managers on election processes, and reviewed documentation of contracts and state statues for vendor compliance. Verification of documentation from voter registration forms to voter recount process.

Group Systems Project, QA Architect,
Responsible for interviewing development team, with an emphasis on developing an QA team to function in an agile development environment.

- Created quality assurance reports
- Trained development team on writing test case and discovering and registering defects
- Established traceability for the development process

Jeppesen JAD Project, Senior Project Manager,
The Jeppesen JAD Automated Tools Implementation Project for the installation of Rational Robot and Rational Test Manager to Jeppesen specifications through the RUP methodology of Inception, Elaboration, Construction, and Transition. Duties included:

- Project team management
- Project responsibility and oversight
- Facilitating and coordinating project meetings

GRP Quality Group, LLC
9/2003 – 11/2005
Quality Architect Consultant at WebMD

- Hired to analyze and improve quality processes
- Interviewed employees and customers regarding their experience and perception of the WebMD Research and Development process
- Compiled a Red Flags document from the interview results
- Established a Quality Map to show present quality state and map the future direction needed to improve quality processes
- Wrote a Project plan using MS Project to show how to achieve their quality goals and presented findings to the Board of Directors

Quality Architect Consultant and Acting Quality Assurance Director for Visa

- Served as Acting Quality Assurance Director for the ROL Project.

- Deliverables include: Test Plan, Best Practice Documentation, Milestone charts & project plan, Daily QA Reports, and Findings reports.
- Lead 30 testing engineers as Director through completion of the Visa Resolution OnLine 5.1 project.
- Implement standards around test documentation, communication, naming conventions, quality gates and metrics.
- Assist in the implementation of TestDirector and QTP.
- Provide test process and methodology expertise
- Design test automation requirements and mentor team on best practices.
- Consult in product, process and people recommendation implementation within Inovant/Visa.
- Created reporting charts for Testing Activities through TestDirector

Project Lead at GE Access

- Test Lead for the Oracle R11i business applications and an Oracle 11.5.9i upgrade project
- Directed sixteen business users testing effort during the R11i 11.5.4 to 11.5.9 upgrade
- Responsibilities included, setting processes, facilitating meetings to report progress, addressing issues, and setting goals
- Reported to the QA Manager on the daily progress

Consultant Project Manager for the GE Access Quality Assurance Automated Tools

As Team Manager on the GE Implementation Project duties included:

- Project oversight
- Facilitating meetings with management
- Project management documentation and reporting regarding the installation of the Mercury Interactive tools suite, - Quick Test Pro, LoadRunner, and TestDirector – through, the scripting and successful execution of automated test cases for existing GE Access.

Deliverables included:

- Team interviews and training, Installation of the Mercury Test Suite, 60% automated test case coverage, Best Practice documentation, Project Charter, Project Plan, Risk Management Plan, Milestone Charts, Detailed External Labor Budget, Communication Plan, Documentation Plan, and a Completion/Transition Plan.

Gartner, Inc.

3/1998 – 9/2003

Director of Quality Assurance for Enterprise Projects

- Assigned to the insure Quality Assurance for all client facing Gartner web sites
- Quality Assurance project planning for Gartner.com (Gartner's Group's primary revenue generating tool)
- Events Registration (the highest single component for revenue collection for Gartner - $100M+ - t

- Worked with 10 reports and the Gartner IS3 group to sync Gartner.com events registration and billing with the internal billing database. CyberSource tools were used for verification of credit cards and checks.
- Managed 4 QA analysts on the GartnerG2.com project, an agile methodology Quality Assurance project
- Responsible for the quality and delivery of various Gartner mini-sites such as Gartner508.com and Marketview.com
- Provided research deliverables through CD-ROM, Lotus Notes, and iWeb (client intranets) were also included under my jurisdiction.
- Maintained a leadership role on the Technical Review Board and in the Technical Leadership Meetings.

Senior Quality Assurance Analyst

- Lead QA on the World Class Web project for Gartner in cooperation with Razorfish, Inc, this project redefined Gartner's web presence and strategy
- Wrote extensive test plans
- Project planning and management
- Supervised 7 to 10 associates in Stamford, CT, and Maitland, FL.
- Achieved the goal of producing an industry leading Web presence
- Produced, assisted, and mentored team in preparing test data, manual testing, automated script debugging, and procedure documentation for the Internet site testing
- Verified data integrity, performance, and scalability at the system level across multiple hardware, software browser and OS platforms and configurations. This included Intel/Windows, MAC/OS10, HP Unix, and Sun/Solaris running various versions of IE, Netscape, Mozella, and Opera.
- Achieve, applied knowledge of HTML, Perl, Java, C++, and Interwoven, and iPlanet technologies, along with a solid working knowledge of Oracle 9

Westbrook Technologies, Inc.
2/1996 – 3/1998
Quality Assurance Manager

- Responsible for the testing aspects of mid and high-end document management applications, which front end to a variety of SQL compliant databases in standalone, network and Internet environments.
- Supervised 5 quality assurance technicians.
- Provided for the successful testing and release cycle of eight major products in two years, through organization and project management skills
- Wrote and supervised the implementation of extensive test plans
- Prepared technical specifications
- Supervised the QA reviews of technical documents.
- Interfaced with customers, sales representatives and software tool kit vendors on special projects.
- Implemented automated software testing and defect tracking methodologies - Segue QA Partner for workstation & server testing, Segue Silk for Internet application testing.

Micro House International
1994-1996
Testing and Support Manager
Micro House International developed leading edge OEM utility software for hard drive vendors such as Maxtor, IBM and Connor/Seagate, and published CD-ROM technical libraries.

- Managed both quality assurance and product support departments (5 staff members). Prepared and implemented extensive test plans. Documented test results for ISO 9000 customers.
- Interfaced with customers on support, product specification and joint QA acceptance processes.
- Established a state-of-the-art testing lab. Implemented training courses in Quality Assurance.

Education & Certifications

- BS in Psychology, Southern Oregon University, 1975
- BS in Sociology, University of Colorado, 1986
- Mercury Interactive Test Director
- WinRunner, Loadrunner
- Watchfire Linkbot
- LogiGear Web Testing

Professional Development

- Six Sigma training;
- Silver Award – GartnerG2.com;
- Bronze Award – Gartner.com

Bruce Geller

Title: Senior Source Code Reviewer

Software Engineer with over 12 years experience in the design, development, testing and maintenance of software applications. Has participated in the design and programming applications for various industries. Experience in functional testing and debugging; Code review for logical function and adherence to standards, database and test case design, and graphical user interface design and implementation.

Technical Experience

C++, C, Pascal, Java 2 SE, Swing, JFC, JBuilder, Visual J++, Visual C++, CodeWarrior, Borland C++, Delphi, SQL, HTML, XML, XSLT, Visual Basic, MFC/WFC, Windows 95/NT/XP, Btrieve, Paradox, InterBase, SourceSafe, PVCS.

Professional Experience

SysTest Labs
Denver, CO
7/2003 – Present

Voting Systems Independent Test Authority (ITA) and Voting Systems Test Laboratory (VSTL), Test Engineer, 7/2003 – Present

Participates in manual and automated voting systems source code reviews to verify that the software conforms to the Federal Election Commission Voting System Standards or EAC Voluntary Voting System Guidelines. Interprets the published Standards and participates in determining their application to voting source code subject to review. Participates in discussions with the software development vendors to communicate the causes and severities of the discrepancies. This includes systems for the following manufacturers:

- Election System & Software
- Populex
- Sequoia Voting Systems
- AutoMARK Techincal Systems
- Hart InterCivic
- Diebold Election Systems
- Accupoll
- Danaher Controls
- NEDAP

Ohio Job Insurance IV&V, Senior IV&V Engineer, 1/2007-5/2007

Provided the IV&V activities and assessments associated with the project areas of Internal Testing of Automated Financial Controls and Data Validation Program Verification. Responsibilities included assessing existing controls and system edit checks to determine whether sufficient checks and balances exist; verification that the files created by the data programs are accurate and correct; as well as conform to IT industry practices.

University of Colorado: CU Connect, Senior IV&V Engineer, 5/2006 – 1/2007

Performed automated performance test planning and test execution of the CU Portal used by 25000+ users. The test cases were scripted in Segue Silk Performer 7.3.1 and executed to evaluate the performance of each channel and changed channel prior to production deployment. Made recommendations to improve the automated testing process and procedures in the existing overall quality assurance context.

ResortQuest, Test Engineer, 10/2005-2/2006

Performance test and analysis efforts for web based enterprise applications and client/server applications. Designed test cases and developed the Segue Software SilkPerformer automated performance test scripts and associated test data. Executed the performance test cases, also provided analysis of results and performance tuning support.

SprintPCS, Senior Test Engineer, 9/2004 – 3/2005

Provided performance testing and analysis. Designed test cases, developed the Segue Software SilkPerformer automated performance test scripts and associated test data. Executed the performance test cases, provided analysis of results, and performance tuning support for an enterprise critical web based application.

Sustain Technologies, Test Engineer, 6/2004-9/2004

Provided performance testing of the case management system for the client. Responsible for planning, designing, developing, and executing automated test scripts using Segue's Silk Performer. This effort included the development of 21 unique test scripts replayed in a Windows 2000 environment.

Colorado Department of Labor Employment, genesis Unemployment Insurance IV&V Project, Senior QA Engineer, 7/2003 – 9/2004

Performed independent verification and validation of the Contractor's deliverables. Assessed the Database Design for both the logical data model and physical data tables. Assessed the Interface Control and H/W and S/W System Documents and compared to requirements. Provided IV&V Code and Peer Review Support and participated in Code and Peer Reviews. Identified software technical issues that resulted in a rework of the software to increase the reliability and maintainability.

Quark, Inc.
Denver, Colorado
Software Engineer 2 5/2000 - 11/2001

Designed and implemented the client-side application for Quark, Inc.'s common license control using JBuilder, Java 2 SE and Swing. The license control client application is the front end of a 3-tiered license control architecture that uses CORBA for data transport. The license control client is distributed with all of Quark, Inc.'s products. Maintained several C and C++ libraries used throughout Quark, Inc.'s products to display cross-platform (Windows and Mac) user dialogs. Isolated and extracted common functionalities from the body of QuarkXPress code for inclusion in new libraries. Reviewed code changes from Quark's India offices for correctness prior to their

integration into QuarkXPress and into the source code archive. Merged and synchronized source code file versions within theSourceSafe code archive.

Ferrell Ventures, Inc.
Lakewood, Colorado
Programmer, 4/1999 - 4/2000
Developed business management applications for the construction industry using MS Visual J++ 6.0 and MS SQL Server 7.0. Participated in application design meetings. Developed graphical user interface prototypes for management approval. Designed database table structures and relationships to support project requirements. Implemented graphical user interfaces. Projects included a license control application to limit the number of concurrent users of other licensed Ferrell Ventures applications.

Software 4 Retail Solutions
Lakewood, Colorado
Software Engineer, 5/1997 - 4/1999

Designed and developed back office applications for the retail grocery industry using Delphi and Visual C++. Consulted with clients to help determine and document user requirements. Developed user interface prototypes for management and client approvals. Designed and implemented database structures to support project requirements. Implemented graphical user interfaces using Delphi and object oriented Pascal. Designed class objects to provide access to the database. Used Visual C++ to implement database access objects as dynamically linked libraries. Projects included an application to automate the retail pricing function for Nash Finch, the nation's third largest grocery products distributor.

Boise State University
Boise, Idaho
Programmer/Analyst, 10/1996 - 5/1997

Programmed under contract with the Federal Highway Administration in Lakewood, Colorado using Delphi. Implemented motor carrier safety inspection software. Developed user interface prototypes for agency approval. Implemented new functionalities and integrated them into the existing Delphi application. The application was distributed to law enforcement agencies throughout both the United States and Canada.

Tradeware Technologies, Inc.
Golden, Colorado
Programmer, 9/1992-10/1996

Programmed point of sale and back office management applications for the fast food, concessions, and entertainment industries using Borland C++ and Visual Basic. Documented client requirements. Developed user interface prototypes for client approval. Designed database table structures and relationships to meet project needs. Researched source code to determine the scope of client requested modifications.

Projects included inventory control, receiving, and payroll applications for use by clients that included United Artists Theatre Circuit, Au Bon Pain, and Arby's Restaurants.

DryWall Supply, Inc., Trans Pacific Stores, Ltd., United Artists Cable
Denver, Colorado
Assistant Controller, Staff Accountant 1980-1991

Reported to Vice Presidential and Controller levels. Responsibilities included G/L, P/R, A/P, A/R, account reconciliations, financial statements, sales tax reports, staff supervision, and development of accounting policies and procedures.

Education

- **Metropolitan State College of Denver,** B.S., Computer Science, 1992
  Elected to The Gold Key Honor Society

- **University of Colorado**, B.A., Biology, 1978
  Cum Laude with Departmental Honors
  Elected to Phi Beta Kappa Honor Society

- **University of Colorado at Denver,** and Arapahoe Community College
  Additional course work in Accounting and Computer Science

Dan McNamee

Title: Senior Automation Specialist
Mr. McNamee has over ten years of experience as a Senior Automation Specialist with expertise in several automated load and regression test tools available on the market, as well as, software development, testing/QA and release management.

Experienced in leading test efforts, writing test documentation, test cases, test scripts, check lists, and defect reports. Knowledgeable in Functional Testing skill set for e-commerce & client/server interfaces.

Mr. McNamee is qualified in Independent Verification & Validation (IV&V) of Hardware and Software Installation

- Monitor and assess the Database Design for both the logical data model and physical data tables; attend Design Reviews and Walkthroughs
- Monitor and assess the Database Build and Test activities, attend Build Walkthroughs and Test Sessions
- Monitor and assess the Architecture Build and Test activities, attend Build Walkthroughs and Test Sessions
- Assess the Logical Data Model and compare to the Requirements and CDLE Data Models and Data Dictionaries.
- Assess Physical Data tables and compare to logical data model.
- Assess the Interface Control Document and H/W and S/W System Document and compare to requirements.

IV&V of the System Architecture Integration

- Monitor and assess Deployment, Implementation, and Maintenance Walkthroughs
- Monitor, assess and ensure that all hardware and support software (operating systems, tools, utilities, etc.) have been installed, setup, and configured according to procedures and requirements.
- Review and assess the Deployment, Implementation, and Maintenance Plans and Maintenance Manuals for the system architecture.
- Review and assess COTS recommendations and warranty information.

IV&V Code and Peer Review Support

- Participated in Code and Peer Reviews in accordance with the ATOM methodology.
- Identified software technical issues that resulted in a rework of the software to increase the reliability and maintainability.

Technical Experience

Hardware Platforms:
IBM PS2 Compatible Hardware, Apple Macintosh, Windows 95/98/NT/2000/XP
Applications:
Siebel, Curam
Software/Languages:
Visual Basic, C/ C++, Java, XML,
Databases:
MS SQL Server 6.5 & 7, Oracle8i, MySQL,

Management, Development and Test Tools:
SilkPerformer, SilkTest, Rational Robot, SunOS4, Solaris, WebTest, WinRunner, LoadRunner, shell, perl

Project Experience

SysTest Labs
Denver, Colorado
1/2002 – Present
McKesson Automation Project, Automation Specialist 10/2007- Present
Lead Automator using WinRunner on a CCMS PowerBuilder application. Mr. McNamee devised a copy written framework approach to increase the effectiveness of automated testing. He leads a team of three (3) automation specialists to successful test the CCMS project. This immense product had not been end-to-end in over 5 years due to time constraints. Mr. McNamee's team's effort will allow testing of the product in less than two weeks. Previously the end-to-end effort was 640 man-hours.

IHS Inerdeq, Senior Automation Specialist, 9/2006 – 10/2006
Performed load testing against the Inerdeq web application. Testing was performed using Compuware's QA Load and Segue's SilkPerformer load testing tools. Performed several tests multiple times against multiple versions of the Inerdeq software and multiple versions of the database to assist with application tuning and defect repair.

Policy Studies Inc., Senior Automation Specialist, 8/ 2006
Performed analysis and discovery on PSI's applications to determine which applications would be candidates for functional and performance automated testing. Provided recommendations for methods of automation and recommendation of automation tools.

BMC Software Remedy Project Senior Automation Specialist, 7/2006
Performed automated testing against the Remedy web application to capture and report response times of the application for system tuning; and performed tests remotely and onsite to assist in diagnosing where possible bottlenecks may lie.
IBM, Tiffany.com, Senior Automation Specialist, 5/ 2006 – 6/2006
Performed load testing for the redesigned Tiffany.com web site. Tests were run in both Rational Performance Tester and in Segue Silk Test. Performed scripting, execution and system setup and maintenance of the entire load generating hardware.
Strongwood Holdings Project, Senior Automation Specialist, 1/ 2006 – 4/ 2006
Creation of GUI tests for the Strongwood web application. Created a script to automate the testing of the application. Tests were driven by datapools so that creating a datapool to perform the test could test all functionality of the application.

Louisiana ACESS Project Senior Automation Specialist, 1/2004 –3/2006
As part of SysTest Labs' Automation team on Louisiana Department of Social Services, ACESS IV&V project Mr. McNamee was responsible for the creation of GUI tests for the ACESS web application. He created a new scripting method for web applications, based on datapools to be more flexible with the application layout, data requirements and execution path to minimize coding changes in the script due to application changes.

Veteran's Administration, MedVA Senior Automation Specialist, 8/2003 – 1/ 2004
Participated in the Veteran's Administration Eligibility Database data conversion effort by developing a Visual Basic application for data validation. This process used the output from the legacy system to build test scripts that performed a compare record by record from the data converted to the new system against the legacy extract

Sprint PCS, Senior Automation Specialist, 1/2003 – 8/ 2003
Mr. McNamee used WinRunner to implement automation framework, library and scripts on Sprint's various web sites. Implemented a Java test driver to test Sprint PCS external SMS API calls on the Server side. DB Visualier was used to verify data from oracle database, and Setup Tomcat, Openware's Handset UP Simulator 4.1 to work with SprintPCS Messaging APIs. Used XML Spy to implement XML test files based on the given DTD.
Used a TestDirector COM object (in C++) to remotely execute Test Sets in TestDirector 7.5. Installed & configured SQL Server. Implemented a QA Web Site using DreamWeaver UltraDev4. The Web Site now has the ability to view, edit, delete and search testcases from a SQL Server. Used TestDirector 7.6 to access and implement test plans and test cases.

SysTest Labs Internal Operations Senior Automation Specialist, 2/2002 – 3/2003
As one of the senior members of the automation test team, Mr. McNamee was responsible for design, development and implementation of WinRunner automated test libraries for use in testing standard interfaces to web based applications being developed in Web Logic, iPlanet, Web Sphere, Apache and others.

Best Buy.com Senior Automation Specialist, 2/2002 – 3/ 2003
Responsible for planning, designing and developing automated regression test scripts using IBM/Rational's Test Studio (Robot). In addition, provided support for the design and development of both Rational Performance Studio and Segue Software SilkPerformer automated virtual user scripts.

Oticon, Senior Automation Specialist, 12/2002 – 3/2003
Responsible for planning, designing, developing, and executing automated regression test scripts using IBM/Rational's Test Studio (Robot). This effort included the development of 7 unique test scripts that had to be replayed in a Citrix environment.

SUSTAIN Technologies, Senior Automation Specialist, 7/2002 –2/2003
Responsible for planning, designing, developing, and executing automated regression test scripts using IBM/Rational's Test Studio (Robot). This effort included the development of 21 unique test scripts that had to be replayed in both a Windows 2000 and Citrix environment.

Seurat – Home Shopping Network, Senior Automation Specialist, 3/2002 –6/2002
Responsible for planning, designing, developing, and executing automated regression test scripts using IBM/Rational's Test Studio (Robot). This effort included the development of 12 unique test scripts that had to be replayed in both a Windows 2000 and Citrix environment.

McKesson
Broomfield, Colorado
1999-2002
Programmer, 2000-2002
Responsible for maintaining and adding functionality to Autobook, a program used internally as well as sold commercially, written in Visual Basic using MSSQL Server and MS Access. My next assignment was maintaining and adding functionality to Algo Editor, a program used internally only, written in C++, Java, XML, and Visual Basic.

SR. QA Analyst, 1999-2000
First position with McKesson was as a SR QA analyst working on maintaining as well as creating new automated test scripts for Autobook using Segue's Silk test. Performed stress/load tests as well as performance tests on Autobook using the automated tests.

Oppenheimer Funds, 1999
Mr. McNamee wrote test plans and performed manual testing on off the shelf products to verify Year 2000 compliance. Products tested include SQL Server, ProComm and Netscape Navigator.

Kinetra LLC,
Denver, Colorado
1999
Test Engineer
Wrote test plans, executed manual tests and created automated tests using WinRunner and WebTest. Testing was performed on both Client/Server applications as well as Web based applications.

Manugistics,
Denver, Colorado
1998-1999
Test Engineer
Wrote and executed test scripts for the AMS (Advanced Manufacturing and Scheduling) software created by Manugistics, as well as creating and updating test plans and other related documentation. I performed manual tests as well as automated tests using the QA Partner testing software. Testing was performed on Windows NT systems.

Renaissance Worldwide, Inc.
Des Moines, Iowa
1998
Test Engineer
Contractor at client, Norwest Mortgage to test applications for year 2000 acceptance. Performed functional tests on in-house applications both manually and with the WinRunner 2000 testing tool. Testing was performed on Windows 95 systems.

Microware Systems Corporation
Des Moines, Iowa
1996 – 1998
Senior Software Engineer
Wrote and executed test suites for the various software titles created by Microware, as well as creating and updating test plans and other related documentation. In addition to testing, performed integration duties as well as researching hardware and software tools to increase the performance of the department and company. Testing was performed on Windows 95 and NT systems, as well as Sun workstations running SunOS 4.

Atari Corporation
Sunnyvale, California
1988 – 1996
Senior Lead Tester, 1994 - 1996
Performed game and application testing for the Atari TOS based computers, MS DOS based computers and on the Atari Lynx and Jaguar gaming systems. In addition to testing, responsible for mastering all EPROMs, Disks and CDs for both test and final manufacture. Worked on the design on several games, including Alien VS Predator and Tiny Toons Adventures. Finally, worked as an Associate Producer on the Rocky Horror Interactive multimedia title.

Operating System Test Programmer, 1991 - 1994
Created software applications to test all areas of the operating system for the Atari TOS based computers. Additional tasks included applications testing, game testing and programming EPROMs for both the computer and games divisions.

Atari Softsource Administrator, 1990 - 1991
Designed, programmed and maintained Atari Softsource, a database of products available for the Atari TOS line of computers. Duties included database design and programming, data entry and product support.

Technical Support, 1988 - 1990
Provided technical assistance to the consumers of all Atari products in the United States. Coverage included both hardware and software. Support was performed by mail, phone and on the GEnie and CompuServe on-line services.

Education & Certifications

- BS Computer Information Systems, Minor: Communications
  Regis College, Denver, Colorado, 1987
- SysTest Labs' ITA Qualification Program
- SysTest Labs' ATOM™ Methodology