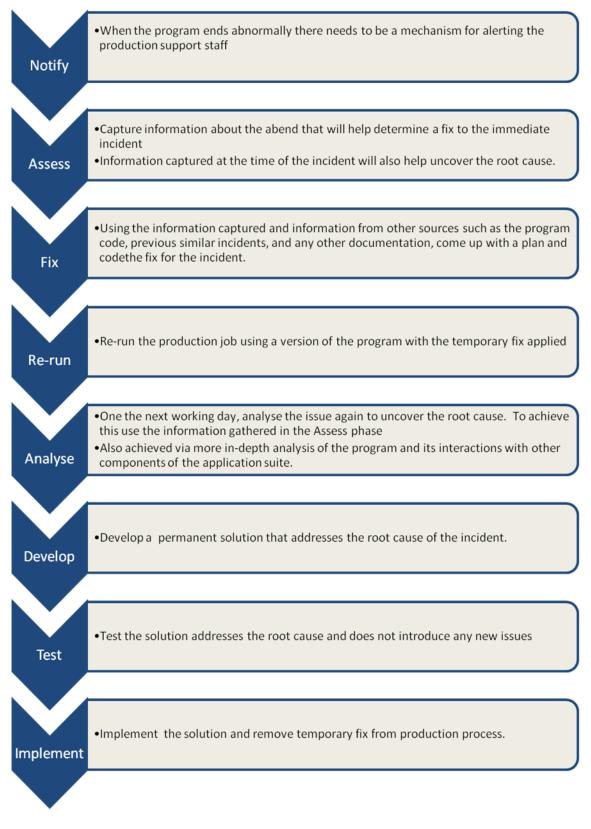
Production Support Use Case

The term Production Support in this context refers to the work done by applications staff to support the successful completion of production batch processing, typically overnight, that forms a vital part of an application. In many sites this batch work must be completed by a certain cut-off time to allow the online system to be fully functional and ready for another day of activity. While there may be a move towards continuous processing and away from the online/batch split, there will always be batch jobs that need to be fixed quickly. Those that support these important batch processes provide a vital service to the organisations that employ them. This chapter shows how the tools and processes of the Optimised Lifecycle can assist Production Support staff address the issues they face.

Production Support Process Flow

Some draw the analogy between the Production Support role in an Information Technology Department and the Triage principles employed in the Emergency Room of a Hospital. While it may not be a matter of life and death the analogy holds true when considering the prioritisation of the work to be done. The issues need to be brought to the attention of those that are best equipped to deal with it. Tasks need to be assessed and prioritised based on their urgency. Jobs on the critical path of the batch stream need to be fixed before jobs that are not. Also it is also important to capture information at the time of the ABEND that will be of value in both dealing with the immediate issue and also later when the root cause of the problem is investigated. Once the immediate concerns are dealt with effectively, the root cause and its solution can be developed, tested and implemented.



(Figure 1)

The Scenario

In this case a batch job has abended and a call has been placed to the on-call production support applications programmer. This is the first of the eight phases that need to occur from this point -

Notify. Notification can be as sophisticated as automation software sending an SMS to on-call staff alerting them to a failure. It could be as simple as a Computer Operator on shift reacting to a non-scrolling message on the system console, issued by the failing job. The main thing is that the notification happens quickly to allow maximum time for the next three steps.

Assess, Fix and Re-run

The Programmer is faced with SOCB ABEND in subprogram CLCLBCST (Figure 2) of TRMTSRCH. The joblog shows that Fault Analyzer has captured the ABEND information. Fault Analyzer lists the following handy information in the job log:

- Program and Module and Source Line number
- ABEND information including a short description (Decimal-Divide Exception).
- The Fault Id and the History File where the abend information can be found

Note: To gain maximum information from the tools insure programs are prepared with the recommended options. Refer to the production documentation for these settings

Edit	: Navigate Search	Project	Run Wind	ow Help												
÷ -	□ ▲ : 参 • (- 0	- : 4	: 🖉 🗸 : 0;			~							F\$.	🔈 Plug-in Deve	əl
			: 🥜	: 🖉 : 🛎												
	JOB04528.out 🔀 🔪															
		J	ES2	JOB L	OG :	SYST	ΕM	MVSE	N O	DEE	OS:	мvзо) 1			1
																-
0	07.13.55 JOB04	528	TUESI	DAY, 22	JUN 2010											
0	07.13.55 JOBO4	528 I	RR010I	USERID RT	POTO8 IS	ASSIGNE	D TO	THIS JOB								
	07.13.55 JOBO4				LAST ACC						2010					
0	07.13.55 JOBO4	528 \$	HASP373	RTPOT08A	STARTED -	INIT 19	-	CLASS A	- SYS MV	3E						
0	07.13.55 JOBO4	528 I	EF403I]	RTPOTOSA -	STARTED	- TIME=0	7.13.	.55								
0	07.13.58 JOBO4	528 -						TIMI	NGS (MIN	3.)			PA	GING COUN	ITS	
	07.13.58 JOBO4			STEPNAME	PROCSTEP	RC	EXCR	CPU		CLOCK	SERV		PAGE	SWAP	VIO SW.	APS
	07.13.58 JOB04		RTPOTOS.		TRTMNT	00	7893		.00		23253	0	0	-	0	C
	07.13.58 JOB04		RTPOTOS.		TRMTSRT	00	206		.00	.00	1745		0	-	0	C
0	07.13.59 JOBO4				halyzer V1						-		-			
(07.14.00 JOB04				RMTSRCH,								(Deci	mal-Divid	le Except:	ior
	07.14.01 JOB04) F01002 a							ST				
0	07.14.01 JOB04	528 I	EF450I I	RTPOTOSA I	RTSRCH -	ABEND=SO	CB UC	0000 REAS	ON=00000	DOB 07	1					
	071			FIME=07.14												
	07.14.01 JOB04		RTPOTOS.		TRTSRCH		1448		.00		11369	0	0	-	0	C
0	07.14.01 JOB04	528 -	RTPOTOS.	A.	TRMTUPDT	FLUSH	C	.00	.00	.00	0	0	0	0	0	C
	07.14.01 JOB04		RTPOTOS.		PATEDIT		C		.00	.00	0	-	0	-	0	C
	07.14.01 JOB04		RTPOTOS.	-	PATDSRT	FLUSH	C		.00	.00	0	-	0	-	0	C
	07.14.01 JOB04	528 -	RTPOTOS.	A.	PATSRCH	FLUSH	0	.00	.00	.00	0	0	0	0	0	0
	<														an a	>
F	Remote Error List 🚺	z/OS Eile	System Mar	pipa 📑 Prop	erty Group Man	ager 🔼 Spi	nnets	Remote S	vstem Details	52			di.		> 🝙 🗦 🏷	~ -
	note system filter My J		- 575com - 10p	ang 1 2 mob	orey areap man	iagor 🛄 ori	ppoto	a romoto a	y scon socalis	-					- - - - - - - - - - -	
	esource		DID	Job Name	Job Owner	Job Entry	F	Return Code	Return Info	System	n ret	User retu	Jr	Return Sta	System Nam	ne :
0	RTPOT08A: JOB04528	JOE	304528	RTPOT08A	RTPOT08	2010/06/2	2 S	OCB	ABENDed				4	ABEND		
	RTPOT08B: JOB04523		304523	RTPOT08B	RTPOT08	2010/06/2			NORMAL			004		COMPLETION		
	RTPOT08D: JOB04522		304522	RTPOT08D	RTPOT08	2010/06/2			NORMAL			000				
	RTPOT08A: JOB04521		304521	RTPOT08A	RTPOT08	2010/06/2			NORMAL			000				
0	RTPOT08A: JOB04519	JOE	304519	RTPOT08A	RTPOT08	2010/06/2			NORMAL			000	(
	RTPOT08B: JOB04517		04517	RTPOT08B	RTPOT08	2010/06/2			NORMAL			004		COMPLETION		
	RTPOT08D:JOB04515		04515	RTPOT08D	RTPOT08	2010/06/2			NORMAL			000		COMPLETION		
	RTPOT08D:JOB04510		04510	RTPOT08D	RTPOT08	2010/06/2			NORMAL			000		COMPLETION		
Ľ.																
<																
															No CICS SM	

(Figure 2)

To view the Fault Analyzer report the programmer should switch to the Fault Analyzer perspective. As detailed in the PD Tools chapter, the history file displayed in the joblog is added via the context menu (Figure 3)

🕏 Fault Analyzer Perspective - J(nal Develope	r for System	Z					_ 0	P D
File Edit Navigate Search Project I	•									
📑 🗉 📄 🗄 🖓 🗸	🌙 🖉 🕈 🖓 🔹 🖄 🗠	{Z - *> ¢	• • • [🐴 Manage Lice	enses			🖹 🖹	ault Analyze	.]
🔍 Fault 🛛 😤 Naviga 🗖 🗖	📄 JOB04528.out 🛛								-	- 6
Set Encoding		JES2	JOB L	0G:	зүзтен	M NVSE	N O	DE EOS	mvso:	^
Delete From View	JOB04528	TUES	DAY. 22	JUN 2010 ·						
Retrieve View Information Retrieve Last Accessed H	1.10804528				ASSIGNED 7	ro THIS JOI	з.			
Add New History File	JOB04528							NE 22, 2010		
	JOB04528		RTPOTO8A			- CLASS A	- SYS MVS	E		
	07.13.55 JOB04528 07.13.58 JOB04528		RTPOTUBA -	STARTED -	- TIME=07.3		INGS (MINS)		
	07.13.58 JOB04528		STEPNAME	PROCETEP	RC E	CP CPU		LOCK SERV	PG P.	
	07.13.58 JOB04528		A	TRTMNT	00 78	393 .00	.00	.05 23253	0	
	07.13.58 JOB04528	-RTPOTO8	A	TRMTSRT	00 2	206 .00	.00	.00 1745	0	
	07.13.59 JOB04528			-				oked by IDI		
	07.14.00 JOB04528				-			# 54: Abend		
	07.14.01 JOB04528 07.14.01 JOB04528				ssigned in ABEND=SOCB	-		NL.V1OR1.HI	ST	
	07.14.01 30804528		TIME=07.14		RREND=SOCR	UUUUU REA:	000000000000000000000000000000000000000	0B 071		
	07.14.01 JOB04528			TRTSRCH	*SOCB 14	48 .00	.00	.03 11369	0	
	07.14.01 JOB04528	-RTPOTO8	A	TRMTUPDT		0.00	.00	.00 0	0	
	07.14.01 JOB04528	-RTPOTO8	A	PATEDIT	FLUSH	0.00	.00	.00 0	0	
	07.14.01 JOB04528			PATDSRT	FLUSH	0.00	.00	.00 0	-	~
	07 14 01 TORO4520	DTDATAG	· A	n Areneu	PT TICU	0 00	10			
E Outline 🕱 🔲 Proper 🗋 🗖	🕄 Default 🚯 Lookup 🔏 F)omoto Sustem [otaile M					8 4 4		-
	Remote system filter My Jobs	Centoce bysceni L					(L)**			
	Resource	Job ID	Job Name	Job Owner	Job Entry	Return Code	Return Info	System ret	User retur	. 1
	RTPOT08A: JOB04528	JOB04528	RTPOT08A	RTPOT08	2010/06/22		ABENDed			
	RTPOT08B: JOB04523	JOB04523	RTPOT08B	RTPOT08	2010/06/22	U0004	NORMAL		004	
	RTPOT08D: JOB04522	JOB04522	RTPOT08D	RTPOT08	2010/06/22	U0000	NORMAL		000	
	RTPOT08A: JOB04521	JOB04521	RTPOT08A	RTPOT08	2010/06/22		NORMAL		000	
	RTPOT08A: JOB04519	JOB04519	RTPOT08A	RTPOT08	2010/06/22		NORMAL		000	
	RTPOT08B: JOB04517 RTPOT08D: JOB04515	JOB04517 JOB04515	RTPOT08B RTPOT08D	RTPOT08 RTPOT08	2010/06/22		NORMAL		004 000	
	RTPOT08D:JOB04515	JOB04515 JOB04510	RTPOT08D	RTPOT08	2010/06/22		NORMAL		000	
					2010/00/22111					
	<									

(Figure 3)

The Fault Analyzer Report (Figure 4) will show the source line that immediately preceded the failure. In this case the divisor in the Compute statement on line 54, PATIENT-COPAY has a value of 0. Hovering over PATIENT-COPAY in RDz would tell the programmer the field is part of the linkage section of the program so is passed to the program from the calling program – in this case TRMTSRCH.

and the second	A/DallasMainframe/FAULTANL.V10R1.HIST/F01002/F01002.far - IBM Rational Developer for System	m z	
File Edit Navigate Search Project	Run Window Help		
i 📬 • 📰 👜 i 💁 •	i 🥖 i 🔗 🔹 i 🖢 - 🖗 - 🏷 🗢 🗢 - i 🔒 Manage Licenses	EP 🗄	Kault Analyze
🕰 Fault 🛛 🛣 Naviga 🗖 🗖	☐ JOB04528.out		- 8
Q FA Artifacts DallasMainframe de Q History Files L Q FAULTANL.V10R1.H	Source Line # 000054 COMPUTE TEMP-COST = 000055 (VENIPUNCTURE-COSTS + ANCILLARY-COSTS + 000056 (PROCEDURE-BASE-COST * 2.2)) 000057 * (REIMBURSE-PCT / PATIENT-COPAY). The COBOL source code for data fields involved in the failure:		~
Cutline Proper Main Report Summary Synopsis Event Summary Ferent 1	Source Line # 00032 05 PATIENT-COPAY PIC 9(3) COMP-3. 000033 05 REIMBURSE-PCT PIC 9(3) COMP-3. 000034 05 PROCEDURE-BASE-COST PIC 9(7)V99 COMP-3. 000035 05 ANCILLARY-COSTS PIC 9(5)V99 COMP-3. 000036 05 VENIPUNCTURE-COSTS PIC 9(5)V99 COMP-3. Data field values at time of abend: ANCILLARY-COSTS = 0.00 PATIENT-COPAY = 0 *** Cause of error *** PROCEDURE-BASE-COST = 211.10 REIMBURSE-PCT = 80 VENIPUNCTURE-COSTS = 326.76 Man Report Event Summary Abend Job Information System Wide Information Misc Information		~
Event 2 Abend Job Information System Wide Information Misc Information	Default Default	Column Co	nfiguration 🗸 🗖 🗖
	Fault_ID Job/Tran User_ID Sys/Job Abend I_Abend F01002 RTPOT08A RTPOT08 ESYSMVS S0CB S0CB	Job_ID JOB04528	Jobname A RTPOT08A V
: D*	Refreshing the followIST(F01001)		No CICS SM connectio

(Figure 4)

The programmer feels they can insert code into the program to check if the field is 0 and move a valid value to it. The programmer consults the application documentation to discover default copayment factor is 100 but that some states will have a lower factor set. The programmer decides to add a test for the field being zero and if so move 100 to the field to eliminate the SOCB.

The code is added and the program is recompiled to a temporary library. The production job is rerun successfully from the temporary library. This allows down-stream processing to complete within the window allowed for batch processing under the service level agreement in place.

```
100-CALC-LAB-COSTS.
** Emergency Fix to allow batch to complete
IF PATIENT-COPAY = ZERO
MOVE +100 TO PATIENT-COPAY.
COMPUTE TEMP-COST =
   ( VENIPUNCTURE-COSTS + ANCILLARY-COSTS +
   ( PROCEDURE-BASE-COST * 2.2 ) )
   * (REIMBURSE-PCT / PATIENT-COPAY ).
```

Analyse Root Cause and Develop Solution

At the next opportunity an analyst within the applications area supporting the batch job, needs to determine the root cause. This involves trying to understand how and what sets the contents of the PATIENT-COPAY field within the patient master file. RAA can be used to determine where a field is

modified within a set of application components including programs and copybooks. Also RAA can be integrated with RDz so that the searches can be done from within an RDz perspective. Right click on the project that contains the elements that are to be searched. Then from within the search window click on the RAA Integration Tab (Figure 5). Enter the field name, PATIENT-COPAY in the search string, specify to search for a Data Element and limit the search to modifications of the data element.

😏 z/OS Projects - IBM Rational Developer for Syst	tem : 😨 Search							_ 7 🗙
File Edit Navigate Search Project <	Customize	Limit To Declaration Reference	: ns () Modificatio	A Integration			asMainframe 2/OS UNIX Files 2/OS UNIX Shells MYS Files My Data Set My Test Files My Test Files My Test Files My Search Q My Search Q TSO Commands JES	Projects **
	Remote system filter My Jobs						8 🗇	
An outline is not available.	Resource	Job ID	Job Name	Job Owner	Job Entry		Return Info	System ret
	RTPOT08A: JOB04528	JOB04528	RTPOT08A	RTPOT08	2010/06/22		ABENDed	
	RTPOT08B:JOB04523	JOB04523	RTPOT08B	RTPOT08	2010/06/22		NORMAL	
	RTPOT08D: JOB04522	JOB04522	RTPOT08D	RTPOT08	2010/06/22		NORMAL	
	RTPOT08A: JOB04521	JOB04521	RTPOT08A	RTPOT08	2010/06/22	U0000	NORMAL	
	RTPOT08A: JOB04519	JOB04519	RTPOT08A	RTPOT08	2010/06/22		NORMAL	
	RTPOT08B:JOB04517	JOB04517	RTPOT08B	RTPOT08	2010/06/22		NORMAL	
	RTPOT08D: JOB04515	JOB04515	RTPOT08D	RTPOT08	2010/06/22	U0000	NORMAL	
	RTPOT08D: JOB04510	JOB04510	RTPOT08D	RTPOT08	2010/06/22	U0000	NORMAL	
i □ ◆			1				🗧 🔻 No CIO	IS SM connection

(Figure 5)

The search reveals a number of references. Each reference needs to be investigated. A modification within PATSRCH shows that another field called COPAY from the PATIENT-MASTER-REC (Patient Master File) is used to populate PATIENT-COPAY.

```
300-CALC-EQUIP-COSTS.

MOVE "300-CALC-EQUIP-COSTS" TO PARA-NAME.

MOVE PATIENT-ID IN INPATIENT-DAILY-REC TO

PATIENT-KEY,

PATIENT-ID IN CALC-COSTS-REC.

READ PATMSTR INTO PATIENT-MASTER-REC.

MOVE COPAY TO PATIENT-COPAY.
```

The task now is to trace how COPAY in PATIENT-MASTER-REC is populated within the entire application. Again the integration between RDz and RAA (RAAi) can be used to complete an

intelligent search of the application. In a similar manner to the search above for PATIENT-COPAY, the analyst searches for modifications of COPAY. This reveals it is set to STATE-FACTOR which in turn is based on an evaluation of the EMP-STATE field (Figure 6).

e Edit Navigate Search Project Ru	Window Help	
** • 🖪 🗁 🖗 🗇 🖗 🖉 🔹	🗛 🕴 🏇 🔹 🔕 🔹 🍕 🗄 🚀 🔹 🖢 😓 🗧 🖓 🔹 🖗 😓 🖓 🔹	😭 🔂 z/OS Projects
🛢 z/OS Projects 🛿 🛛 📄 🍸 🗖	🗖 🖨 CALCCOST.dbi 📄 MSTRUPDT.dbi 🛛 🦳 🗖 🗄	🗆 📕 Remo 🖾 \Upsilon 🗖
Chapter 3 .project	Line 625 Column 62 Insert -1-B-++2+3+45+6	7
🗉 🧰 BuildOutput	EVALUATE EMP-STATE	
BABENDREC.cdv	WHEN "NC" MOVE 100 TO STATE-FACTOR WHEN "NJ" MOVE 100 TO STATE-FACTOR	□ 🔁
	WHEN "NOVE 100 TO STATE-FACTOR	😥 💒 New Connection
BNCHDALY.cpy	WHEN "ND" MOVE 60 TO STATE-FACTOR	😟 📑 Local
BNCHINS.cpy	WHEN "AZ" MOVE 100 TO STATE-FACTOR	😑 ᆒ DallasMainframe
BNCHMRKQ.jd	WHEN "AR" MOVE 75 TO STATE-FACTOR	🚊 💱 z/OS UNIX F
	WHEN "ID" NOVE 100 TO STATE-FACTOR	z/os unix s
BNCHPRSN.cpy	WHEN "DE" MOVE 80 TO STATE-FACTOR	🖃 🍾 MVS Files
BNCHTRMT.cpy	WHEN "WA" MOVE 100 TO STATE-FACTOR	😟 🦆 My Data
CALCCOST.cbl	WHEN "TX" MOVE 100 TO STATE-FACTOR	E AREFERE
CLCLBCST.cbl	WHEN "PA" MOVE 90 TO STATE-FACTOR	E 🔆 My Tesl
COBPERF.cpy	WHEN "HI" MOVE 100 TO STATE-FACTOR	E A Refere
CUSTREC.cpy	WHEN "CA" MOVE 99 TO STATE-FACTOR	E B2 LIB
CUSVSREC.cpy	WHEN "OR" MOVE SO TO STATE-FACTOR	IIII → DE2 LIB
DALYEDIT.cbl	END-EVALUATE	
		🚽 🤿 My Sear
DATACALC.cpy	COMPUTE PATIENT-TOT-AMT =	TSO Comma
	(WS-LAB-CHARGES + WS-EQUIP-CHARGES)	🖻 😽 JES
Properties 🗄 Outline 🛛 👘	* ((REIMBURSE-PCT / 100) + (STATE-FACTOR / 100))	
Jaz 🖨 🖬 🔹 📽		
2200-EXIT.	▲ ⇒ MOVE STATE-FACTOR TO COPAY IN PATIENT-MASTER-REC.	
2400-GET-EQUIP-PROVIDER.		
- 2400-EXIT.		
	S STATE-FACTOR PIC 999 VALUE 0.	
3000-EXIT.		
4000-CALCULATE-EQUIPMENT-C		
4000-EXIT.	🐼 Remote Error List 🕀 z/OS File System Mapping 🖳 Property Group Manager 🔛 Snippets 🚂 Remote System Details	Search S?
	🚱 Remote Error List 🥂 2/OS Hie System Mapping 🙀 Property Group Manager 🛅 Shippets 🛃 Remote System Details	A bearen w
5000-EXIT.		🖻 🕺 🔳 🍢 🕶 🗋
6000-COMPUTE-IN-NETWORK.	COPAY: 4 Data Element Occurrences Found	
6000-EXIT.	■ De Chapter 3	
7000-EXIT.	MSTRUPDT.cbl (4 Matches)	
1000-DB2-ERROR-RTN.	Instruction of reductions)	

(Figure 6)

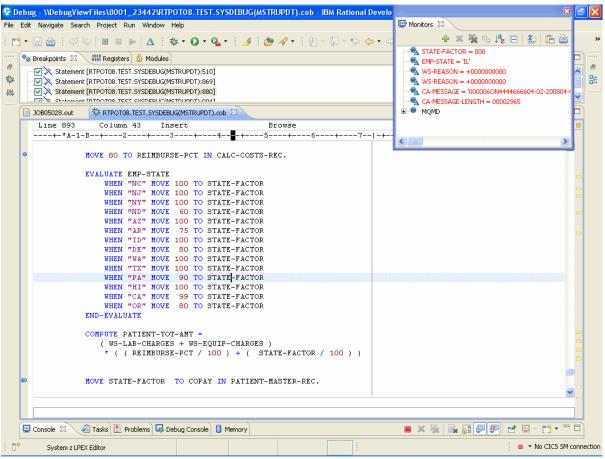
The analyst notices two things from looking at the code excerpt and from hovering over the field:

- The list of State codes is not exhaustive and there is no WHEN OTHER clause on the Evaluate to catch unexpected values.
- The initial value of STATE-FACTOR is set to 0 in the working-storage.

This means that if a State code other than those listed explicitly in the Evaluate statement, COPAY will be left at the initial value of 0 rather than the accepted default value of 100. To test the theory that there are values of EMP-STATE possible that are not catered for, RDz and its Debug perspective are used. Once in the debug session the following is done to setup for the test:

- A breakpoint is set at the following statement: MOVE STATE-FACTOR TO COPAY IN PATIENT-MASTER-REC.
- Highlight the fields we are interested in (EMP-STATE and STATE-FACTOR), right click and choose to monitor them. In this way we can see their values change as the program is debugged.

Figure 7 shows the results after a couple of records are processed in the debug session. Note the values of EMP-STATE and STATE-FACTOR in the Monitor window.



(Figure 7)

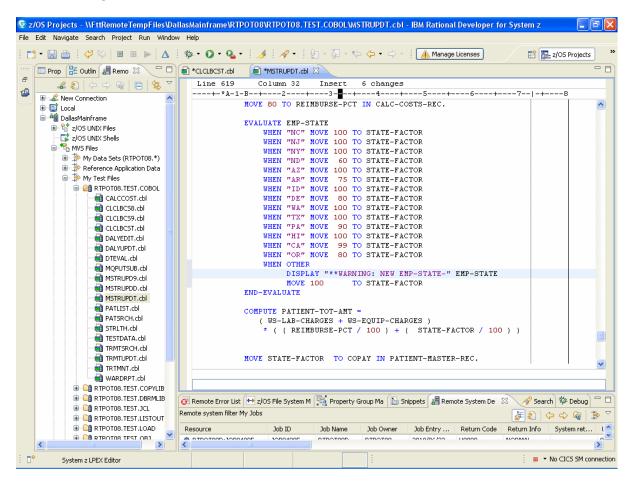
EMP-STATE had a value of "IL". This is not catered for in the Evaluate statement so at the breakpoint, just before COPAY in PATIENT-MASTER-REC is modified, STATE-FACTOR is still at its initial and invalid value of 0.

It is good practice with EVALUATE statements to always include a WHEN OTHER clause. This makes the program more resilient. While normally the addition of new states from a business perspective may have meant the applications area would be notified so they could make the necessary program changes, this does not always happen. Programs need to be written to expect the unexpected.

The initial intention of the application was to have a default Co-Payment factor of 100 but to allow for lower factors for some states. The change shown in Figure 8 is agreed upon. The change will set the factor to the default value of 100 but also display a warning message on the joblog. It may be better if these exceptions were also written to an error file but for this example we will use a display to the job log only. The idea is to highlight the issue so it can be followed up and addressed, rather than letting it keep occurring and causing unforeseen issues as it did in this scenario.

The fix will have three effects. Firstly the original intention to have a default factor set will be implemented and secondly it will stop any future repetition of this SOCB abend. Thirdly, the warning message will highlight to the applications team that a new state is now incorporated into the database. This will provide an opportunity to discuss the introduction of the new state with a

business representative and ensure that the co-payment factor chosen for each is appropriate rather than defaulting.



(Figure 8)

Test and Implement Solution

Implementing the solution involves following the development and testing methodology adopted by the site in question. Examples of the methodologies are discussed earlier in Section IV, including: traditional approaches such as the waterfall method; collaborative approaches such as the Iterative Approach using RUP or SCRUM; Agile Methodology which can enabled by use of a collaboration tool such as Relational Team Concert for z (RTCz) (Figure 9 & 10). RTCz like RAA can be integrated with RDz and forms an important part of IBM's Enterprise Modernization Workbench.

🚱 Defect 370: RTPOT08A - S0CB in CLCLB	CST - IBM Rat	
National Team Conce	rt.	John Milne 🔻 Log Out 🕐 🗸
Dashboards Project Areas Work	Items Plans Source Control Builds Reports	EM Workbench Redbook 🗸
₽ .	🖲 Defect 370	🛱 😹 🍇 🔗 🛛 Save
Create	Summary: * RTPOT08A - S0CB in CLCLBCST = New	✓
Work Item	Overview Links Approvals History	Saved: Jul 10, 2010 10:47 A.M.
	Details	Quick Information
Recently Viewed	Type: Defect Owned By: Jon Sayles	Subscribers (1): JM
Defect 370	Severity: 1 Major Viority: 2 Medium V	
Shared Queries Predefined Pr	Found In: Unassigned Planned For: Sprint 1 (1.0) Creation Date: Jul 10, 2010 10:48 A.M. Estimate: Correction: Created By: John Milne Time Remaining:	
	Description Edit	
	overnight master update suite abended in step TRTSRCh. S0CB ABEND in subprogram CLCLBCST of TRMTSRCH.	

(Figure 9)

🚱 Defect 370; RTPOT08A - SOCB in CLCLB	IST - IBM Rat		
Closed subscribed by me * P New unassigned	Tags:		~
	Description	Edit	
	overnight master update suite abended in step TRTSRCh. SOCE of TRMTSRCH.	BABEND in subprogram CLCLBCST	
	Discussion (2 comment)	Add Comment	
	Collapse All Expand All		
	1. John Milne Jul 10, 2010 10:51 A.M.		
<	Fault Analyzer report indicates the divisor in the Compute s has a value of 0. I had a look at the doco and it suggests th For now I will add a check to see if PATIENT-COPAY is 0 ar job. We need someone to look into this tomorrow to see wi	e default value of this should be 100. Id if so set it to 100. Then re-run the	
	2. John Milne Jul 10, 2010 10:58 A.M.		
	following code added to 100-CALC-LAB-COSTS. ** Emergency Fix to allow batch to complete IF PATIENT-COPAY = ZERO MOVE +100 TO PATIENT-COPAY. job rerun		
		Add Comment	
		Save	
IBM.		da22	~

(Figure 10)

Irrespective of the approach, it is important to test the changes have the desired results. After retrieving the programs, the changes are made in a test environment. The job is run in that test environment using sample data and/or copies of production data. As expected it reveals a number of states that have been introduced to the database without changes to MSTRUPDT to ensure the COPAY field is correctly set (Figure 11).

There may be more testing required before the program is migrated to production, including a system/integration test of the whole system to ensure this change does not have unforeseen consequences downstream. The other task that needs to be done at the same time the solution to the root cause is migrated to production, is to remove the temporary fix to a version of CLCLBCST to ensure the new, full solution is picked up when the suite next runs.

Edit Navigate Search Project	o <mark>ut - IBM Rational Develo</mark> t Run Window Help	per for Syste	m z						_	لحال
1 • □ ≜ 救 • Q • (· 如 - 和 -	*	🗸 🕴 👍 Mar	hage Licenses			E	🛃 z/OS Projec	ts
R ⊠ ² □ □	JOB05065.out	OB05069.out	JOB0507	3.out 🛛						
A New Connection A New Connection	**WARNING: NEW EI **WARNING: NEW EI	MP-STATE-IN MP-STATE-C' MP-STATE-C' MP-STATE-C' MP-STATE-C' MP-STATE-IN MP-STATE-IN MP-STATE-IN MP-STATE-IN MP-STATE-IN MP-STATE-IN MP-STATE-IN MP-STATE-C' MP-STATE-C' MP-STATE-C' MP-STATE-C' MP-STATE-IN MP-ST		*******		s (JB Remote S	vstem Details 5	3 Search	h) ∰⊳ Debug]	
	Remote system filter My Jobs								(+ +) (Q	→
RBN REE	Resource RTPOT08A: JOB05073	Job ID	Job Name RTPOT08A	Job Owner RTPOT08	Job Entry	Return Code U0000	Return Info NORMAL	System ret	User retur	R
	 RTPOT08A: JOB05073 RTPOT08D: JOB05071 	JOB05073 JOB05071	RTPOT08A RTPOT08D	RTPOT08	2010/06/22	U0000	NORMAL		000	c
🔜 ner 🎽		30005070	RTPOTODD	DTDOTOO	2010/00/22	10000	HORPHAL		000	>
<										



Summary

This chapter has demonstrated how the tools of the Optimised Lifecycle can be used to improve the response to a production support ABEND. It is possible to look into these sorts of issues without the tools discussed and without abend analyzers such as Fault Analyzer, or third party alternatives. COBOL can provide a symbolic dump and if a compile listing, with appropriate options, has been kept it is possible to derive the failing line of code from standard module/offset ABEND information. Also applications staff with deep application knowledge and experience can quickly drill-down on issues without much help. In fact this is how production support issues were investigated and may still be at some sites.

What was demonstrated in this chapter was how the tools now available and the integration between them can benefit those who tackle these issues on a daily basis. These Production Support incidents are often a stressful events for this involved that, if not addressed in a timely manner, can have major negative impacts to the organisations concerned. The value of the tools is that they can reclaim time that in the past is burned identifying the point of failure and indentifying a fix. This time saved can be the difference between getting systems back on track or not.