

> Effective requirements analysis and test design using decision tables

-Ramesh Ramani

Presented on:18th September 2007



Agenda

- > Key challenges
- > Root cause
- > Introduction to decision table
- > Case study
- > How to derive a decision table
- > Conclusion



Key challenge



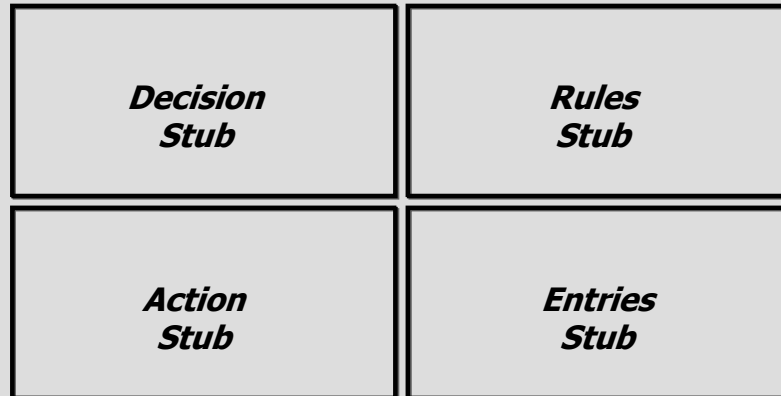
- > Requirements
 - Not understanding the requirements completely
 - Requirement gaps
 - Assumption
- > Test design
 - Time spend on requirements review and test design being high
 - No covering the requirements completely
 - Not having adequate coverage for negative and invalid combinations

Root Cause

- > Ambiguity in requirement
- > Incomplete requirement
- > Plain English

Decision Table Layout

- > Standard format used for presenting decision tables.



Case study-Plain English

- > Acme Outdoor World has just announced that it will offer free shipping for all orders over \$250.00 (Applies to standard shipping for 4 or more items totaling \$250.00 or more)
- >). Shipping charges for all other orders will be prorated.
- > For orders less than \$250.00, the shipping charge will be calculated as follows:
- > If the number of items is 3 or less:

Delivery	Shipping
Next Day	\$35.00
2nd day	\$15.00
Standard	\$10.00

Case study-Plain english

- > If the number of items is 4 or more:

Delivery D	Shipping Charge, N = number of items
Next Day	N * \$7.50
2nd day	N * \$3.50
Standard	N * \$2.5

- > For orders over \$250.00, the shipping charge will be calculated as follows:
- > If the number of items is 3 or less:

Delivery	Shipping
Next Day	\$25.00
2nd day	\$10.00
Standard	N*\$1.5.00



7

Case study-plain english

- > If the number of items is 4 or more:

Delivery D	Shipping Charge, N = number of items
Next Day	N * \$6.00
2nd day	N * \$2.50
Standard	Free



8

Case study- Pseudo code

```
> If purchase amt. > $250.00
>   If number of items < 4 then
>     If delivery date is next day then
>       Delivery charge = $25.00
>     Else if delivery date is 2nd day then
>       Delivery charge is $10.00
>     Else if delivery date is standard then
>       Delivery charge is $1.50 per item.
>     End If
>   Else
>     If delivery date is next day then
>       Delivery charge is $6.00 per item
>     Else if delivery date is 2nd day then
>       Delivery charge is $2.50 per item
>     Else
>       Delivery is free of charge
>     End If
>   End If
```

Case study- Pseudo code

```
> Else
>   If number of items < 4 then
>     If delivery date is next day then
>       Delivery charge = $35.00
>     Else if delivery date is 2nd day then
>       Delivery charge is $15.00
>     Else if delivery date is standard then
>       Delivery charge is $10.00
>     End If
>   Else
>     If delivery date is next day then
>       Delivery charge is $7.5 per item
>     Else if delivery date is 2nd day then
>       Delivery charge is $3.50 per item
>     Else
>       Delivery is $2.50 per item
>     End If
>   End If
```

Decision table equivalent

	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8	SC9	SC10	SC11	SC12
Purchase Amount > \$250	Y	Y	Y	Y	Y	Y	N	N	N	N	N	N
Number of Items is 4 or More	N	N	N	Y	Y	Y	N	N	N	Y	Y	Y
Delivery	Next	2nd	Std	Next	2nd	Std	Next	2nd	Std	Next	2nd	Std
Shipping Charge (\$)	25	10	N*\$1.5	N*\$6	N*\$2.5	free	35	15	10	N*\$7.5	N*\$3.5	N*\$2.5

OR

Purchase amount	Over \$250						Less than \$250					
Number of Items	3 or less			4 or more			3 or less			4 or more		
Delivery day	Next	2nd day	Std	Next	2nd day	Std	Next	2nd day	Std	Next	2nd day	Std
Shipping Charge (\$)	25	10	N*\$1.5	N*\$6	N*\$2.5	free	35	15	10	N*\$7.5	N*\$3.5	N*\$2.5

Steps to Create a decision table

1. List all causes in the decision table
2. Calculate the number of possible combinations
3. Fill columns with all possible combinations
4. Reduce test combinations
5. Check covered combinations
6. Add effects to the table



Simple example

- > A store wishes to program a decision on non-cash receipts for goods into their intelligent tills. The conditions to check are agreed as:
 - > Transaction under £50
 - > Pays by cheque with cheque card (guarantee £50)
 - > Pays by credit card
- > The possible actions that a cashier could take are agreed as:
 - > Ring up sale
 - > Check credit card from local database
 - > Call a supervisor
 - > Automatic check of credit card company database
- > Using the rules above construct a decision table showing all possible combinations of alternatives.

Step 1: List all causes

Cause	Values
Under £50	Y,N
Pays by cheque	Y,N
Pays by credit card	Y,N
Effect	
Ring up sale	
Check from local database	
Call Supervisor	
Check credit card database	

Hints:

- > Write down the values the cause/condition can assume
- > Cluster related causes
- > Put the most dominating cause first
- > Put multi valued causes last

Step 2: Calculate combinations

Number of Values to the power of the number of causes with these values



- > If all causes are simply Y/N values:
 $2^{\text{number of causes}}$
- > If 1 cause with 3 values and 2 with 2:
 $3^1 * 2^2 = 12$
- > Or, use the Values column and multiply each value down the column, eg.
 $3*2*2=12$
- > In the above example we are going to use the first formula $2^3 = 8$

Step 3: Fill columns

Cause	Values	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8
Under £50	Y,N	Y	Y	Y	Y	N	N	N	N
Pays by cheque	Y,N	Y	Y	N	N	Y	Y	N	N
Pays by credit card	Y,N	Y	N	Y	N	Y	N	Y	N
Effect									
Ring up sale									
Check from local database									
Call Supervisor									
Check credit card database									

> **Algorithm:**

- ★ Determine Repeating Factor (RF): divide remaining combinations by the number of possible values for that cause
- ★ Write RF times the first value, then RF times the next etc. until row is full
- ★ Next row, go to 1.

Step 4: Reduce combinations

Cause	Values	SC1	SC2	SC3	SC4	SC5	SC6	SC7	SC8
Under £50	Y,N	Y	Y	Y	Y	N	N	N	N
Pays by cheque	Y,N	Y	Y	N	N	Y	Y	N	N
Pays by credit card	Y,N	Y	N	Y	N	Y	N	Y	N
Effect									
Ring up sale									
Check from local database									
Call Supervisor									
Check credit card database									

- > Find indifferent combinations – place a '-' or colour them differently
- > Join columns where columns are identical
- > Tip: ensure the effects are the same

We can see that some of the condition rules are invalid, the customer cannot pay by cheque AND pay by credit card or not pay by either method. We have decided that these combinations are mutually exclusive. This decision table can be reduced to 4 condition rules.

Step 5: Check covered combinations

Cause	Values	SC2	SC3	SC6	SC7
Under £50	Y,N	Y	Y	N	N
Pays by cheque	Y,N	Y	N	Y	N
Pays by credit card	Y,N	N	Y	N	Y
Effect					
Ring up sale					
Check from local database					
Call Supervisor					
Check credit card database					

- > Checksum
- > For each column calculate the combinations it represents
- > Add the number of valid and invalid scenarios the values should be equal to the number of scenarios = $2 \times 2 \times 2 = 8$

Step 6: Add effects to table

Cause	Values	SC2	SC3	SC6	SC7
Under £50	Y,N	Y	Y	N	N
Pays by cheque	Y,N	Y	N	Y	N
Pays by credit card	Y,N	N	Y	N	Y
Effect					
Ring up sale		X			
Check from local database			X		
Call Supervisor				X	
Check credit card database					X

> Read column by column and determine the effects

> One effect can occur in multiple test combinations



Summary



Advantages	Disadvantages
<ul style="list-style-type: none"> • Easy to manage the change and clear communication between stake holders • Takes less time to create test scenarios and test data from the requirement 	<ul style="list-style-type: none"> • Need to understand the basics of decision table and spend initial hrs in learning
<ul style="list-style-type: none"> • Tests provide effective examples for validating required expressions 	<ul style="list-style-type: none"> • Requires understanding of Boolean Algebra
<ul style="list-style-type: none"> • Smoothly incorporates dependency information and automatically avoids "impossible" interpretations 	<ul style="list-style-type: none"> • Requires automated support
<ul style="list-style-type: none"> • Provides a mechanism for verifying dependency information 	<ul style="list-style-type: none"> • Requires learning and experience

