DECISIONS, DECISIONS – HOW TO USE DECISION TABLES FOR EFFECTIVE TEST DESIGN

RANDALL W. RICE, CTAL-FULL, CTFL-AT, CMT
WWW.RICECONSULTING.COM

AGENDA

- The main concept of decision tables
- How to create simple decision tables
- How to create decision tables with invalid dependencies
- Why and how to reduce decision tables
- How decision tables help in test design
- How decision tables help in requirements analysis
- Tool support for creating decision tables
- Explanation of ASTQB Foundation Level Sample Question #24
- Additional Resources

DECISION TABLE TESTING

- **Decision tables are a good way to capture system requirements that contain logical conditions, and to document internal system design.**
- **They may be used to record complex business rules that a system is to implement.**
- **Can be applied as part of both black-box and white-box test design techniques.**

3

RICECONSULTING

KEY POINT

- **You do not need complete definition of a requirement to apply decision tables.**
- **That is the beauty of the technique.**
  - Tell me one rule and I can deduce the rest.
- **That's why this is also great for requirements analysis!**
  - How many times have you seen "gaps" in requirements where certain combinations of conditions or events were not considered?

4

RICECONSULTING

DECISION TABLE TESTING (2)

- **A logical way to derive test cases**
- **Best applied with a limited number of rules**
  - (7 rules with T/F decisions yields 128 possible test cases. 8 rules – 256 cases, etc.)
  - The formula is 2 to the nth power, where n is the number of conditions
- **Each rule can be seen as:**
  - T : True
  - F : False
  - I : (Invalid, illogical)

5

RICECONSULTING

"SIMPLE" EXAMPLE

- **If an employee is hourly and works over 40 hours in a weekly pay period, they are paid overtime for each hour worked over 40 hours. Overtime pay is 1.5 times the regular hourly pay.**

Step 2: Define the T/F (or Y/N) values. Tip: Do the "happy path" first, then vary the values.

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Employee is hourly	T	T	F	F
Over 40 hours worked in a week	T	F	F	T
Actions				
Overtime pay (1.5 x) for each hour worked > 40.	T			
No overtime pay		T	T	T

Step 1: Define the conditions

Step 3: Define the actions (outcomes)

Each "rule" is a test case.

6

RICECONSULTING

IDENTIFYING ILLOGICAL CONDITIONS

- **Notice that in Rules 2, 3 and 4, the outcome is the same – No overtime paid.**
- **However, in Rules 3 and 4, we really don't care if the person works over 40 hours or not. They aren't eligible for OT anyway.**

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Employee is hourly	T	T	F	F
Over 40 hours worked in a week	T	F	I	I
Actions				
Overtime pay (1.5 x) for each hour worked > 40.	T			
No overtime pay		T	T	T

7

RICECONSULTING

REDUCING THE TABLE

- **We can consider the condition of working over 40 hours in a week invalid or illogical if the employee is not hourly.**
- **Now, notice that Rules 3 and 4 are identical, both in conditions and outcome.**
- **So we only need 3 Rules or test cases for full decision table coverage.**
- **However, you may choose to perform all 4 cases with T/F values only as negative tests for cases 3 and 4.**

	Rule 1	Rule 2	Rule 3	Rule 4
Conditions				
Employee is hourly	T	T	F	F
Over 40 hours worked in a week	T	F	T	F
Actions				
Overtime pay (1.5 x) for each hour worked > 40.	T			
No overtime pay		T	T	T

8

RICECONSULTING

WHY REDUCE A TABLE?

- **To achieve more efficiency in test design.**
  - Fewer cases, no loss in condition coverage.
- **However, as we just mentioned you may want to test all combinations of conditions, including the "invalid" ones, as negative tests.**
  - There may be cases where the invalid cases are impossible to achieve.
  - Example: Testing the IE browser on a Mac O/S.

9

RICECONSULTING

MORE COMPLEX DECISION TABLE EXAMPLE

- **Rule 1 to be tested says:**
  - If all of the following are true:
    - Employee is hourly
    - Hours worked in a pay period is over 40
    - No sick time, vacation or holiday time accrued during the week in excess of OT
    - No holidays worked during the week
  - Then the following actions take place
    - Pay is computed as 40 hours at the employee's regular pay rate
    - Overtime pay for each hour over 40
    - Overtime hourly rate is 1.5 times employee's regular rate.

10

RICECONSULTING

DECISION TABLE EXAMPLE (2)

		Rule 1	Rule 2	Rule 3	Rule 4
Condition	Employee is hourly	T	T	T	T
	Hours worked in a pay period is over 40	T	T	T	F
	Sick time, vacation or holiday taken < OT hours	T	T	F	I
	No holidays worked during week	T	F	I	I
Actions	Hours up to 40 computed at regular pay, Hours over 40, less sick, vacation and holiday hours, computed at 1.5 times regular rate	X	X		
	Holiday pay computed for hours worked on the holiday at 2 times regular rate		X		
	No overtime pay rate applied			X	X

11

RICE CONSULTING

HOW DECISION TABLES HELP IN TEST CASE DESIGN

- **They identify logical variations**
- **They identify variations that do not need to be tested because of illogical combinations**
- **They help determine test case completeness**

12

RICE CONSULTING

HOW MUCH TO COVER?

- The coverage standard commonly used with decision table testing is to have at least one test per column, which typically involves covering all combinations of triggering conditions.

13

RICECONSULTING

WHITE-BOX EXAMPLE – MULTIPLE CONDITION COVERAGE

IF (A > B) or ((C < D) and (A<B)) THEN
PERFORM X
ELSE
PERFORM Y;

1. Create decision table
2. Eliminate invalid or duplicate cases
3. Convert to test cases

Note the test cases are rows in this example instead of columns

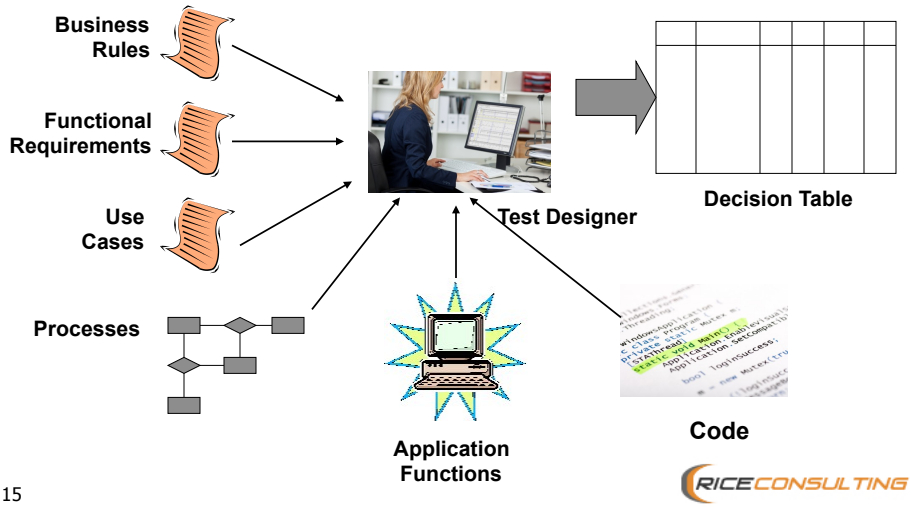
A>B	C<D	A<B	O/C
~~T~~	~~T~~	~~T~~	~~X~~
T	T	F	X
~~T~~	~~F~~	~~T~~	~~X~~
T	F	F	X
F	T	T	X
F	T	F	Y
F	F	T	Y
F	F	F	Y

A	B	C	D	Expected
6	3	2	4	Perform X
6	3	4	2	Perform X
3	6	2	4	Perform X
1	1	2	4	Perform Y
3	6	4	2	Perform Y
1	1	4	2	Perform Y

14

RICECONSULTING

HOW DECISION TABLES ARE CREATED



15

RICECONSULTING

TOOLS

- This is often a manual effort
- However, a great tool for automating the work is Richard Bender's BenderRBT tool.
  - http://www.benderrbt.com

16

RICECONSULTING

FINALLY…INFAMOUS QUESTION #24 FROM THE ASTQB SAMPLE EXAM

#24 You have been given the following conditions and results from those condition combinations. You can only have one form of payment. A PIN is only needed for a debit card. Given this information, using the decision table technique, what is the minimum number of test cases you would need to test these conditions?

Conditions:
Valid cash
Valid credit card
Valid debit card
Valid PIN
Bank accepts
Valid Selection
Item in Stock
Results:
Reject Cash
Reject Card
Error Message
Return Cash
Refund Card
Sell Item

a. 7
b. 13
c. 15
d. 18

17

RICECONSULTING

MY SOLUTION APPROACH

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Conditions:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	Valid cash	T	T	T	F	I	I	I	I							
3	Valid credit card	I	I	I	I	T	T	T	T	F						
4	Valid debit card	I	I	I	I	I	I				T	T	T	T	F	T
5	Valid PIN	I	I	I	I	I	I				T	F	T	T	I	T
6	Bank accepts	I	I	I	I	T	F	T	T	F	T	F	T	T	F	F
7	Valid Selection	T	T	F	I	T	T	F	F	I	T	I	T	F	I	
8	Item in Stock	T	F	T	I	T	T	T	F	I	T	I	F	I	I	
9	Results:															
10	Reject Cash				T											
11	Reject Card						T									
12	Error Message		T	T	T		T									
13	Return Cash		T	T												
14	Refund Card															
15	Sell Item	T	F	F	F	T	F				T					

For the purposes of answering the exam question, I do not spend time on defining the results.

18

RICECONSULTING

ASTQB SOLUTION

#24
C is correct. See table.

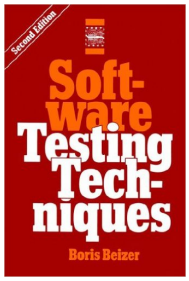
Conditions:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Valid cash	Y	Y	Y	N												
Valid credit card					Y	Y	Y	Y	N							
Valid debit card										Y	Y	Y	Y	Y	N	
Valid PIN										Y	N	Y	Y	Y		
Bank accepts					Y	N	Y	Y		Y		N	Y	Y		
Valid Selection	Y	N	Y		Y		N	Y		Y			N	Y		
Item in Stock	Y		N		Y			N		Y				N		
Results:																
Reject Cash				Y												
Reject Card						Y			Y		Y	Y			Y	
Error Message		Y	Y	Y			Y	Y					Y	Y		
Return Cash		Y	Y													
Refund Card							Y	Y					Y	Y		
Sell Item	Y				Y					Y						

DECISION TABLE SUMMARY

- **The value of decision table testing is that it creates combinations of conditions that might not otherwise have been identified during test design or exercised during testing.**
- **It may be applied to all situations when the action of the software depends on several logical decisions.**

FURTHER REFERENCES

- **The Art of Software Testing by Glenford Myers**
  - He ties together decision tables with cause-effect graphing.
- **Software Testing Techniques, 2nd Ed. By Boris Beizer**
  - Very complete treatment of the topic.
- www.abebooks.com
- www.alibris.com



21

RICECONSULTING

YOUR QUESTIONS?

22

RICECONSULTING

23

RICECONSULTING

BIO - RANDALL W. RICE

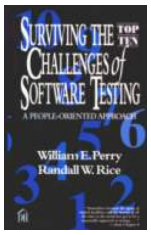
- Over 35 years experience in building and testing information systems in a variety of industries and technical environments
- ISTQB Certified Tester – Foundation level (CTFL), Advanced Level (CTAL) Full
- ASTQB Certified Mobile Tester (CMT)
- ISTQB Foundation Level Agile Tester (CTFL-AT)
- Director, American Software Testing Qualification Board (ASTQB)
- Chairperson, 1995 - 2000 QAI's annual software testing conference
- Co-author with William E. Perry, ***Surviving the Top Ten Challenges of Software Testing*** and ***Testing Dirty Systems***
- Principal Consultant and Trainer, Rice Consulting Services, Inc.

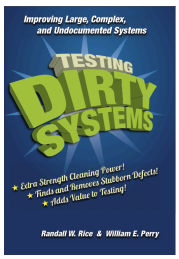
24

RICECONSULTING

CONTACT INFORMATION

Randall W. Rice, CTAL
Rice Consulting Services, Inc.
1608 SW 113th Pl
Oklahoma City, OK 73170
Ph: 405-691-8075
Fax: 405-691-1441
Web site: www.riceconsulting.com
e-mail: rrice@riceconsulting.com





25

RICECONSULTING