Anuruddha Jayasekara Pathiranage

02/12/2015

Decision table testing

A decision table is an outstanding technique used in both testing and requirements management. It is a structured exercise to prepare requirements when dealing with complex business rules. Decision Table Testing is useful to deal with a combination of inputs, which produce different results. It helps reduce test effort in verifying each and every combinations of test data, at the same time ensuring complete coverage. Decision tables are very much helpful for testers to search the effects of combinations of different inputs and other software states that must correctly implement business rules(Chilenski & Miller, 1994).

Example of Decision Table Testing

Pearson A is new customer and he needs to open new account for shopping card. Then there are three conditions. If he is new, he will get a 12% discount on all his purchases today, second if he is an existing customer and he holds a loyalty card, he gets a 8% discount and third if he has a rewards , he can get 15% off today (but it can't be used with the 'new customer' discount). Discount amounts are added, if applicable.

Conditions	**Rule 1**	**Rule 2**	**Rule 3**	**Rule 4**	**Rule 5**	**Rule 6**	**Rule 7**	**Rule 8**
New customer (12%)	**T**	**T**	**T**	**T**	**F**	**F**	**F**	**F**
Loyalty card (8%)	**T**	**T**	**F**	**F**	**T**	**T**	**F**	**F**
Rewards (15%)	**T**	**F**	**T**	**F**	**T**	**F**	**T**	**F**
Actions								
Discount (%)	**X**	**X**	**15**	**12**	**23**	**8**	**15**	**0**

Table 01. Decision table for shopping card

When Decision Table testing should be used

Decision tables can be used in all situations where the outcome depends on the combinations of different choices, and that is usually very often. It is useful when dealing with complex business rules. In many systems there are tons of business rules where decision tables add a lot of value.

Types of software testing problems that decision table testing addresses

We will apply Equivalence Partitioning and Boundary Value Analysis techniques to only specific conditions or inputs. In contrast to Equivalence Partitioning and Boundary Value Analysis techniques, we use decision table testing for following types of software problems. If we have dissimilar inputs that result in different actions being taken or we have a business rule to test that there are different combination of inputs which result in different actions, we will go for decision table testing, in contrast to Equivalence Partitioning and Boundary Value Analysis techniques("How to use Decision Tables," 2012).

Key assumptions of decision tables testing

When conditions are mutually exclusive, exactly one must be true. Extended entry decision tables typically have mutually exclusive conditions.

Limitations of decision tables testing

1) Total sequence- The total sequence is not clearly shown. No overall picture is given by decision tables as presented by flowcharts.
2) Logic- where the logic of system is simple, flowchart nearly always serve the purpose better than decision tables.
3) Need to decide what conditions are relevant for testing. This may require domain knowledge.
4) Scaling up can be massive. 2^n rules for n conditions that is if the condition are binary. It gets worse if the values are more than binary("What is Decision table in software testing?," n.d.).

References

Chilenski, J. J., & Miller, S. P. (1994). Applicability of modified condition/decision coverage to software testing. *Software Engineering Journal*, *9*(5), 193–200. https://doi.org/10.1049/sej.1994.0025

How to use Decision Tables. (2012, June 25). Retrieved February 13, 2017, from http://reqtest.com/requirements-blog/a-guide-to-using-decision-tables/

What is Decision table in software testing? (n.d.). Retrieved February 13, 2017, from http://istqbexamcertification.com/what-is-decision-table-in-software-testing/