Deploying ISTQB test Framework for DO-278A as an Aviation Safety System testing Standard

Raha Ashrafi, Ramin Nassiri, Sam Jabbedari
Master Student of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran
Computer Engineering Department, Central Tehran Branch, Islamic Azad University, Tehran, Iran
Computer Engineering Department, North Tehran Branch, Islamic Azad University, Tehran, Iran
Raha.ashrafi@yahoo.com, R_Nasiri@iauctb.ac.ir, Sjabbehdari@gmail.com

ABSTRACT

DO-278A is a Well-known FAA CNS/ATM software development standard and ISTQB is a worldwide software testing standard framework. DO-278A verification is mostly done via a variety of methods including Review, Analysis and a few other techniques that are not actually sufficient in testing so it would be quite reasonable to deploy ISTQB techniques for that purpose. This Paper aims to show that Static Techniques of ISTQB may be deployed successfully for DO-278A verification while there are further areas in which it could also be used.

KEYWORDS

Aviation software system, CNS/ATM, DO-278A, ISTQB, Verification.

1 INTRODUCTION

Since the emerging advances in most technologies are much tied to their relevant software, the significant role of software in nearly all aspects of human life is undeniable. That is while in current industries many functions are delegated to software. One of these industries that is affected by software either directly or indirectly is aviation industry as a mission critical example. In this industry, software has a major contribution to safety and air traffic management. Among several factors related to operational equipment in aviation industry, safety has the highest importance because sometimes the costs resulted from lack of safety may not be compostable and/or negotiable.

Once air travel is integrated as a part of human life, in this area safety gets its utmost importance due to high costs that sometimes may not be compensable in comparison with other properties related to the equipment used in this field. Software are being highlighted everyday and there is not any considerable area without software intervention because they play a greatly important role and also because producers have found that the more functions done by softwares the smaller hardwares needed and the lower cost for modification and reproduction relative to modifications and reproduction of hardware. Aviation industry is one of the costliest industries because people lives might be at risk and would be the cost of any mistake. For this reason, this area is a mission-critical and challenging area. Aviation incidents such as Warsaw incident in which software was one of the factors in chain of events indicate the importance of software role in the aviation industry. Therefore, the organizations involved had to step on a path that the ultimate goal of which is protecting people lives and creating a safe condition for aviation industry to meet transportation requirements. FAA was pioneer in this issue and has taken effective steps to develop standard to software.

To meet this purpose, various tests are executed on the software before they become operational. There are several standards for testing. One of them is ISTQB as an international software test framework.

This paper compares DO-278 and FAA standard for producing ground-based software of CNS/ATM, and ISTQB, standard for Software
testing, for finding framework testing according to DO-278 and ISTQB.

2 DO-278A
DO-278A that was published in 2011 is stand-alone standard for producing CNS/ATM ground-based software. However, its previous version, DO-278, is used together with DO-178B which is a standard for airborne software. According to DO-178B, the system failure caused by software errors is divided into five categories. (Table 1)[1]

<table>
<thead>
<tr>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catastrophic</td>
<td>Its costs is not compensable such as human lives</td>
</tr>
<tr>
<td>Hazardous</td>
<td>The situation is difficult to control and can lead to catastrophic if it gets out of control</td>
</tr>
<tr>
<td>Major</td>
<td>The situation is under control but failure to control the situation becomes hazardous</td>
</tr>
<tr>
<td>Minor</td>
<td>The situation is under control</td>
</tr>
<tr>
<td>No Safety effect</td>
<td>A situation that would not affect safety</td>
</tr>
</tbody>
</table>

According to five levels of system failure, software assurance levels are defined. (Table 2)

<table>
<thead>
<tr>
<th>Assurance Level</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>B</td>
<td>Hazardous</td>
</tr>
<tr>
<td>C</td>
<td>Major</td>
</tr>
<tr>
<td>D</td>
<td>Minor</td>
</tr>
<tr>
<td>E</td>
<td>No Safety Effect</td>
</tr>
</tbody>
</table>

It shows that if assurance level is A, the catastrophic incident will not happen or in order to avoid a catastrophic incident, the software should be at level A of assurance.

Table 3 associates DO-278A with DO-178C.

<table>
<thead>
<tr>
<th>Table 3 Assurance Levels association</th>
<th>DO-278A</th>
<th>DO-178C</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL1</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>AL2</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>AL3</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>AL4</td>
<td>Not Equivalent</td>
<td></td>
</tr>
<tr>
<td>AL5</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

AL4 in DO-278A does not have any equivalent in DO-178C, since it was designed for ground-based systems when AL3 is strict and AL5 is too permissive for that.[3]

2.1 Software Life cycle processes

Software life cycle processes in DO-278A are:
-Software planning processes
-Software development processes
-Integral processes

And the software development processes are:
-Software requirements process
-Software design process
-Software coding process
-Integration process

And integral processes are:
-Software verification process
-Software configuration management process
-Software quality assurance
-Approval liaison process

Integral processes are performed concurrently with software planning and development processes throughout software life cycle and following that there is:

![Concurrent Activity Diagram]

Figure 1 Concurrent Activity
In this paper, the review of verification processes is considered.

3 ISTQB

ISTQB is the software-testing standard, which is published in 2002. In this standard, techniques used are divided into two general classes: [4]

- Static
  - Dynamic

3.1 Static techniques

Static category without running Software, detects the error and includes two main techniques:

- Review
  - Static analysis

In which each of them have their relevant sub-branches as follows:

- Review:
  - Informal review
  - Walk through
  - Technical review
  - Inspection

- Static analysis:
  - Data flow
  - Control flow

3.2 Dynamic techniques

Dynamic techniques have three main categories and each of them have their subclasses as follows:

- White box technique
  - Statement Coverage
  - Condition Coverage
  - Decision Coverage
  - Multiple Conditions

- Black box technique
  - Equivalence Partitioning
  - Boundary Value Analysis
  - Decision Table
  - State Transition
  - Comparison

- Experience based technique

4 Review Consideration of DO-278A and ISTQB

Generally, DO-278A verification is done by review, analysis and testing. However, there is not any special technique for verification. Since verification objectives are described in tables of annex A of DO-278A, ISTQB review techniques can be used for DO-278A verification. Corresponding tables of DO-278A objectives and ISTQB techniques are as follows:

<table>
<thead>
<tr>
<th>DO-278</th>
<th>ISTQB</th>
<th>REVIEW</th>
<th>Static Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Informal Review</td>
<td>Walkthrough</td>
</tr>
<tr>
<td>L.L Requirement comply with H.L</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>L.L Requirement are Accurate &amp; Consistent</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>L.L Requirement are compatible with target computer</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>L.L Requirement are Verifiable</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>L.L Requirement Confirm to standard</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
According to objective tables, the following results are derived:

From among ISTQB Static techniques, informal review is not used for DO-278A, because DO-278A is about Safety-Critical software and informal review is not used for safety-critical software then it results in ISTQB Static Review that can be used in DO-278A as follows:

- Walkthrough
- Technical Review
- Inspection
- Data Flow
- Control Flow

**5 Conclusions**

DO-278A is for producing CNS/ATM software and ISTQB is a software testing Framework. By using ISTQB techniques in DO-278A we could be able to achieve a robust testing technique to fulfill testing requirements. ISTQB Static Techniques deployed for CNS/ATM Software are:

- Walkthrough
- Technical Review
- Inspection
- Data Flow
- Control Flow

This paper intends to show the capability of ISTQB Framework to test mission-critical systems as a proof of the compatibility of that framework. There are also further areas, which may be used ISTQB that would be published later on.

**6 REFERENCES**


2. Software Considerations in Airborne Systems and Equipment Certification , RTCA Inc. , 1992
