TITAN

Turnaround **I**ntegration in **T**rajectory **A**nd **N**etwork

Project Number: 233690

Verification Plan

CLASSIFICATION: PU **ISSUE:** 1.0 **DATE:** 16/11/2012

DOCUMENT REFERENCE				
Project	**Work Package**	**Partner**	**Nature**	**Issue**
TITAN	WP 4	RWT	DEL	05

Project co-funded by the European Commission and TITAN consortium.

DOCUMENT CONTROL			
Responsible	**Company**	**Name**	**Date**
Author	RWTH	Athanasios Katsaros	11/01/2012
Partners involved	JEPPESEN	John Butcher, Alicia Grech	
	SLOT	Balázs Kerülő	
Reviewer	JEPPESEN	John Butcher, Alicia Grech	23/05/2012
	INECO	Laura Serrano, Ana Sáez	
Approval	INECO	Laura Serrano	31/05/2012

DOCUMENT CHANGE LOG			
Issue	**Date**	**Author**	**Affected Sections / Comments**
0.1	11/01/2012	Athanasios Katsaros	Creation
0.2	23/01/2012	Athanasios Katsaros	WP4 partners' comments
0.3	14/02/2012	Athanasios Katsaros	Comments in WP4 "Specification and Verification" Teleconference on 31.01.2012 Turnaround Tool Specification Document v0.6
0.4	27/02/2012	Athanasios Katsaros	WP4 partners' and Coordinator's comments
0.5	30/03/2012	Athanasios Katsaros	Comments in 1st WP4 Progress Meeting on 08.03.2012 Turnaround Tool Specification Document v0.7 (Approved version)
0.6	18/05/2012	Athanasios Katsaros	WP4 partners' comments
0.7	31/05/2012	Athanasios Katsaros	WP4 partners' final comments
1.0	16/11/2012	Ana Sáez	Final Version after EC Approval

DOCUMENT DISTRIBUTION		
To/Cc	**Organisation**	**Name**
To	EC	Mr. Remy Denos
To	INECO	Laura Serrano
To	INECO	Ana C. Sáez
To	INECO	Jose Luis Martin Sanchez
To	JEPPESEN	John Butcher
To	JEPPESEN	Alicia Grech
To	SLOT	Roland Gurály
To	SLOT	Balázs Kerülő
To	RWTH Aachen	Athanasios Katsaros
To	RWTH Aachen	Sebastian Kellner
Cc	ISDEFE	Martijn Koolloos
Cc	ISDEFE	Marta Sanchez
Cc	CRIDA	Nicolas Suarez
Cc	CRIDA	Eva Puntero
Cc	BLUSKY	Steve Zerkowitz
Cc	SESAR JU	Paul Adamson
Cc	AENA/SESAR JU	Alejandro Egido
Cc	AENA/SESAR JU	Francisco Javier Fernández de Liger

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

The current document, referred to as “Verification Plan”, builds together with the “Verification Report” the verification activities that aims at the high level TITAN objective of developing a decision support tool for airlines to achieve a more efficient turnaround process by implementing the TITAN concept. Based on “Turnaround Tool Specification Document”, a verification plan is developed to ensure TITAN tool proper implementation by verifying the correct behaviour of its modules/components in a CDM environment against pre-defined requirements.

This document lays down all necessary verification-related information; verification method, verification scenario (conditions), requirements and test (use) cases to be verified and verification conduction.

1. INTRODUCTION

1.1 Document Scope

According to [21] the objective of WP4 is to develop a decision support tool demonstrator for airlines to better evaluate and negotiate any changes in their schedule due to modifications affecting the turnaround process of one or several of their aircraft (to achieve a more efficient turnaround process).

To ensure, that the TITAN component will be implemented properly, the resulting tool should be verified on the basis of the use cases (functional requirements) and non-functional requirements defined in the Turnaround Tool Specification Document [13]. The tasks to be fulfilled in the context of verification are the following:

- create verification plan, aim of this deliverable;
- verify the correct behaviour of TITAN component modules in a CDM environment;
- elaborate verification report.

1.2 Document Structure

The document consists of the following chapters:

- Chapter 1 is the current introductory section.
- Chapter 2 defines how the tool will be verified. After a short but substantial description of the existing verification methods the most appropriate one for our case is chosen and analytically described.
- Chapter 3 gives an answer to the question under which conditions the tool will be verified. The particular scenario that is going to be used for the tool verification is described in full detail filtering out the different milestones and the scenario time sequence.
- Chapter 4 is answering the question what is actually going to be verified. For that reason all functional and non-functional requirements to be tested in the context of particular test (use) cases are analytically presented according to formal guidelines.
- Finally, in chapter 5 the question by whom and when the tool is going to be verified is answered giving information about the conduction of the tool verification.

1.3 Intended audience

This public document (PU) may be distributed freely both within and outside the TITAN consortium. It is primarily a technical document intended to be of use to the verification team that is planned to test the tool/demonstrator after its development and implementation to see if specified requirements are successfully met or not and also to explain the basis of the planned verification process in TITAN to all TITAN partners as well as those outside TITAN who are engaged in verification processes, especially at the airport side.

1.4 Associated documentation

[1] Binder, R. V., 1999. *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Amsterdam: Addison-Wesley Longman Publishing Co.

[2] Butcher, J., 2011. *TITAN WP4 Data Required Document*. 4th Version. Jeppesen.

[3] Butcher, J., 2011. *TITAN WP4 Scoping Document*. 3rd Version. Jeppesen.
[4] Elfriede, D., 2002. *Effective Software Testing: 50 Specific Ways to Improve Your Testing*. 1st ed. Amsterdam: Addison-Wesley Longman Publishing Co.
[5] E-testing Hub, 2007. *Software Testing-Testing Life Cycles*. [online] Available at: <http://www.etestinghub.com/testing_lifecycles.php#2> [Accessed 23.12.2011].
[6] Gurály, R., Kerülő, B., Král, N., 2011. *TITAN WP2 Airport CDM and TITAN - a Comparison*. 1st Version. Slot Consulting.
[7] IEEE, 1991. *IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990*. New York: IEEE.
[8] IEEE Computer Society, 2005. Software Testing. In: IEEE Computer Society, ed. 2004. *Guide to the Software Engineering Body of Knowledge (SWEBOK(R))*. IEEE Computer Society Press.Ch.5.
[9] Jenney, J. et al., 2011. *Modern Methods of Systems Engineering: With an Introduction to Pattern and Model Based Methods*. 1st ed. CreateSpace.
[10] Jungmayr, S., 2003. *Improving testability of object-oriented systems*. PhD. FernUniversität in Hagen.
[11] Kaner, C., 2006. Exploratory Testing. In: QAI (Quality Assurance Institute)*, 26th Worldwide Annual Software Testing Conference*. Orlando, Florida, USA November 2006. Florida: QAI.
[12] Kaner, C., Bach, J. and Pettichord, B., 2001. *Lessons Learned in Software Testing: A Context-Driven Approach*. New York: John Wiley and Sons Inc.
[13] Kerülő, B., 2012. *TITAN WP4 Turnaround Tool Specification Document*. 6th Version. Slot Consulting.
[14] Koolloos, M., 2010. *Guidelines for System Verification (ISGOTI-105245-1LL)*. 1st ed. Madrid: ISDEFE.
[15] Koolloos, M., 2011. *TITAN WP3 Validation Exercise Plan*. 7th Version. ISDEFE.
[16] Laycock, G. T., 1993. *The Theory and Practice of Specification Based Software Testing*. PhD. Department of Computer Science, University of Sheffield.
[17] Marchenko, A., 2007. *XP Practice: Continuous Integration*. [online] Available at: < http://agilesoftwaredevelopment.com/xp/practices/continuous-integration> [Accessed 23.12.2011].
[18] National Aeronautics and Space Administration (NASA), 2007. *NASA Systems Engineering Handbook (NASA/SP-2007-6105)*. 1st Rev. Washington D.C.: NASA.
[19] Pan, J., 1999. *Software Testing*. In: 18-849b Dependable Embedded Systems. [online] Pittsburgh: Carnegie Mellon University. Available at: < http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/> [Accessed 23.12.2011].
[20] Savenkov, R., 2008. *How to Become a Software Tester*. 1st ed. Roman Savenkov Consulting.
[21] Turnaround Integration in Trajectory and Network (TITAN) Collaborative Project, (2011). *TITAN Annex I - "Description of Work"*. 5th ed. Madrid: INECO.
[22] Wanderlei, S. and Reginaldo, A., 2009. Abstract Testability Patterns. In: *Proceedings of the 3rd International Workshop on Software Patterns and Quality*. Orlando, Florida, USA 25 October 2009. [pdf] Available at: <http://se.inf.ethz.ch/old/people/ciupa/papers/esecfse07.pdf> [Accessed 23.12.2011].

1.5 Abbreviations and Acronyms

A	Availability
A/c	Aircraft
(A)CDM	(Airport)Collaborative Decision Making

AD	Administration/Administrator
AIBT	Actual In-Block Time
AIRS	Airport Information Report Service
ALDT	Actual Landing Time
AOBT	Actual Off-Block Time
AOC	Aeronautical / Airline Operations Centre
ARR	Arrival
ASAT	Actual Start-Up Approval Time
ASRF	Aircraft Status Reporting and Forecasting
ASRS	Aircraft Status Reporting Service
ATC	Air Traffic Control
ATFM	Air Traffic Flow Management
ATOT	Actual Take-Off Time
AXIT	Actual Taxi-In Time
AXOT	Actual Taxi-Out Time
BA	Baggage Agent
BFIS	Baggage Flow Information Service
BTL	Baggage Tag List
BW	Blue Wings Airlines
C	Common
CFMU	Central Flow Management Unit
CA	Check-in Agent
CS	Concessionaires
CMMI	Capability Maturity Model Integration
CRUD	Create, Read, Update, Delete functionalities
CTOT	Calculated Take-Off Time
D	Detail Screen View
DEL	Deliverable
EH	Event Handling
EIBT	Estimated In-Block Time
ELDT	Estimated Landing Time
EO	Equipment Operator
EOBT	Estimated Off-Block Time
ETOT	Estimated Take-Off Time

EXIT	Estimated Taxi-In Time
F	Functional
FC	Flight Crew
GH	Ground Handler
GUI	Graphical User Interface
HMI	Human Machine Interface
IC	Information Classification
ID	Identity
LD	Load
M	Milestone
NA	Network Architecture
NF	Non-Functional
OPS	Operations
PAX	Passengers
PBS	Passenger and Baggage Search
PD	Platform Dependency
PFIS	Passenger Flow Information Service
PNL	Passenger Name List
PNR	Passenger Name Record
PSCP	Passenger Security Control Personnel
PU	Public Document
R	Reliability
RET	Rapid Exit Taxiway
RSAT	Requested Start-Up Time
RWY	Runway
S	System or Summary Screen View
SAU	System Access and Use
SDT	System Downtime Testing
S / STND OCC'D	Stand Occupied
TIS	TITAN Information Sharing / System
TITAN	Turnaround Integration in Trajectory and Network
TOBT	Target Off-Block Time
TSAP	Target Start-Up Approval Time
T(U)C	Test (Use) Case

U	Usability
UI	User Interface
VIP	Very Important Person
VTTCS	Variable Taxi Time Calculation Service
WA	Warning
WP	Work Package

2. VERIFICATION METHODOLOGY

2.1 Differentiation between Verification and Validation

System (software) verification and validation activities are highly associated with system (software) engineering, as they are both used to assure that the anticipated system (software) can satisfy its users' needs and requirements. Although quite similar, verification and validation address different issues and have fundamentally different objectives. The difference between those two procedures can be described as follows [7]:

- Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. The question "Have we built the software right and does it match specification?" is answered.
- Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. The question "Have we built the right software and is this what the customer wants?" is answered.

As defined in [21], scope and objective of TITAN verification activities is to verify the TITAN tool to be developed and implemented against specifications set at the beginning of the development phase.

2.2 Methods of System (Software) Performance Verification

Verifying system (software) requirements is a formal engineering process that starts with requirements analysis and ends when the system is accepted by its customer [9]. During system (software) verification steps must be taken to verify that the system (software) satisfies every "shall" statement in the requirements. Verification activities should include in summary:

- Development/implementation of the most appropriate verification strategy/methodology;
- Development/conduction of verification plans;
- Documentation of "shall" requirements verification/compliance (results).

According to [9] and [14] there are four (4) standard verification methods used by systems engineers:

- analysis;
- inspection;
- demonstration; and
- testing - functional (see chapter 2.3.2) or non-functional (see chapter 2.3.3).

Among the existing system (software) verification methods, which are analytically described above, a hybrid form of testing and demonstration seems to be the most appropriate one for the verification of TITAN tool (tool demonstrator (UI) delivered in testable form by the design team) regarding its conformance to required performance, physical characteristics and design construction features set in the Turnaround Tool Specification Document [13]. Required software operability and meeting of predetermined software responses (based on requirements) under specific scenarios will be verified. In case that compliance with requirements is obvious from the design (i.e. system oriented architecture) then design as verification method can also apply [9]. On

the other hand, inspection and analysis are rejected as they would better fit in a case of verifying against more physical characteristics or in the absence of physical software.

2.3 Verification Method - Software Testing

2.3.1 Software Testing - Overview

Software testing is the process of executing a software or application with the intent of finding errors or other defects [11]. The time point at which test effort begins depends on the chosen testing model. According to the **Waterfall Model** testing is performed by an independent group of testers after the functionality is fully developed (requirements have been defined and the coding process has been completed) [5]. On the other hand, the Capability Maturity Model Integration (CMMI) proposes to start software testing at the same moment the project starts and introducing a continuous process until the project finishes [4]. However, emerging agile development models often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers, while software updates are published to the public frequently [17]. Furthermore, depending on development phase different testing levels can be defined [8]; unit/component (1^{st} level) testing; integration (2^{nd} level) testing; system (3^{rd} level) testing; and system integration (4^{th} level) testing.

For TITAN tool verification is decided to be conducted also in parallel with development and implementation process and not only after their completion (system testing). An agile verification method will be implemented constituting of the following elements:

- Reviews or inspections of the software code are to be conducted by the software development team, to which software code remains internal. Such a task does not constitute part of the formal software verification process conducted by the verification team and being described in the current document. (unit/component and integration testing)
- Software executables delivered by the development team to WP4 stakeholders will be verified by the verification team to provide feedback back to the developers' team during software development and implementation ("smoking testing" of tool sub-releases to prevent formal test failure). They do not though constitute part of the final verification process and their results are not planned to be included in the verification document (D4.5).

In the name of such a development of TITAN tool, testing was initiated from the beginning of the development of the tool. So we do not speak in our case for the traditional verification method, in which the testing team verifies the product of development at the very end.

Software testing presupposes software testability that is defined as the degree to which a software artefact supports testing in a given test context [1] [10] [22]. Testability of software requirements is assured when software requirements are consistent, complete, unambiguous, quantitative and verifiable in practice. Through proper requirements setting/definition in Turnaround Tool Specification Document [13] that is reviewed by development and verification team, requirements testability is assured.

Requirement gaps constitute a common defect source. As early error detection can reduce relevant time and money costs, it is better to conduct testing throughout the software life-cycle [12]. For TITAN tool defect prevention is generated in the following stages:

- To prevent tool failure due to false requirements setting the specification document is multiply reviewed and continuously improved before tool design and implementation.

- Prevention of failure due to false tool design and implementation is achieved by development parallel testing (software code reviews or inspections by the development team and smoking testing of software executables – 1st and 2nd level testing).
- System testing once the tool is developed and implemented (3rd level testing).

2.3.2 Functional Testing

There are many approaches to software functional testing. **Static testing** includes reviews, walkthroughs, or inspections, whereas actually executing programmed code with a given set of test cases is referred to as **dynamic testing**. Dynamic testing takes place when the program itself is used for the first time and may begin before the program is 100% complete in order to test particular sections of code.

Regarding access to internal software structure (box approach), software testing can be divided traditionally into the following testing methods:

- **White-Box Testing**

  The tester has access to the internal data structures including the software code.

- **Black-Box Testing**

  The software is treated as a “black box” without any knowledge of internal implementation.

  Specification-based black box testing aims to test the functionality of software according to the applicable requirements [16]. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester (use-cases), who then can simply verify that for a given input the output value (or behaviour) either “is” or “is not” the same as the expected value specified in the test case [20].

In the case of TITAN tool verification, it is decided to implement **dynamic specification-based black box testing**. The tool, once completed, will be tested to verify that the specified requirements are efficiently addressed. The code is considered as a “big black box”; the information can be put into the black box that sends something out. Based on the functional requirements knowledge, we know what to expect the black box to send out and the tests (use cases) to make sure that the black box sends out what is supposed.

2.3.3 Non-functional Testing

In contrast with functional testing, which establishes the correct operation of the software, **non-functional testing** verifies that the software functions properly even when it receives invalid or unexpected inputs. Testing of particular non-functional software requirements includes:

- Performance/stress testing (reliability)
- Usability testing
- Security testing
- Destructive testing (robustness)

In case of TITAN tool verification, specific non-functional requirements are already set during specification documentation. Such non-functional requirements include: network architecture, platform dependency, reliability, availability as well as human-machine-interface (HMI) design. To test the tool against such requirements, non-functional testing would be necessary. In this context

reliability testing, **usability** (or HMI design testing) and **availability**-downtime testing could be implemented.

2.3.4 The testing process

According to the **waterfall development model** a typical cycle for testing includes following phases [19]; requirements analysis, test planning, test development, test execution, test reporting, test result analysis, defect retesting, regression testing and test closure. To the case of TITAN tool verification all above phases up to test reporting apply. Requirements analysis is conducted by specification documentation. The verification plan has to be set, according to which verification will be executed based on test development. After successful verification execution testing results will be reported to the developers' team that will decide how to proceed with possible identified tool defects.

During the verification of TITAN tool following testing products will be used:

- **Requirements verification matrix** [9] [14] [18]

  During system testing steps must be taken to verify that the system satisfies every "shall" statement in the requirements. These shall statement requirements are collected in a document called the Verification Matrix, which is a documentation that defines for each requirement the verification method, the level and type of unit for which the verification is to be performed and any special verification conditions.

- **Test (use) cases**

  A test case normally consists of a unique identifier, requirement references from a design specification, preconditions, events, a series of steps (actions) to follow, input, output, expected result and actual result. A test case is an input and an expected result [8]: "for condition x your derived result is y".

- **Traceability matrix**

  A traceability matrix correlates requirements or design documents to test documents (i.e. requirements to test (use) cases).

- **Requirements compliance matrix**

  The data resulting from the actions summarized in the verification matrix for verifying that the system meets all requirements are collected in a compliance matrix. The compliance matrix shows performance for each requirement. It identifies the source of the performance data and shows if the design is meeting all requirements.

3. VERIFICATION SCENARIO

3.1 Baseline Scenario

TITAN tool verification is based on the "Missing Passenger Scenario" (missing passenger(s) disrupt(s) the end of the turnaround process), which is fully described in [15] (Scenario 1). This scenario is briefly presented as follows:

The aircraft in the scope is a Blue Wings Airlines Airbus A320 at one of their destination airports, 2 hours flying time from their home base. This is the first rotation of the aircraft in the morning. The aircraft is configured for 149 passengers.

The departure airport is in a free-flow situation, however, due to capacity constraints at the destination airport, Blue Wing flights need to obtain a departure slot from the Central Flow Management Unit (CFMU).

Weather is good at both the departure and the destination airports so it is not a factor in the conduct of the flight.

Fuel is cheaper at the departure airport than at the airline's home base and so the aircraft arrived with minimum fuel and was re-fuelled during the turnaround.

The aircraft is parked at a remote position as this is cheaper and the airline follows a kind of low-cost business model. Passengers are brought to the aircraft by bus. This is a Schengen flight (no passport control).

The airport concerned has recently completed a new passenger lounge facility with numerous concessions selling merchandise as well as numerous eateries scattered along the route that passengers are obliged to take from the security checkpoints to the passenger gates of the terminal concerned.

Aircraft at remote stands are served by bus which departs from the ground floor of the terminal and passengers need to use escalators to reach the so called "G" gates. These are similar in layout to the gates with air bridges but the doors open to the tarmac where the buses park with the doors aligned with the building doors. Although the way to these "G" gates is clearly marked, the walking distance is longer and the boarding process must start earlier than in the case of aircraft docked at an air bridge.

(Aircraft parking location determines the passenger gate which in turn impacts on required boarding time.)

The airport has Airport Collaborative Decision Making (A-CDM) tool implemented with all functionalities and additionally TITAN is operational, accessed via end-user applications at all users, including mobile users (de-icing, push-back, passenger transit buses, etc.).

Blue Wings Airlines has a sophisticated Aeronautical Operations Centrer (AOC) at its base airport.

Handling at the outstations is performed by the same multinational company. Both the AOC and the handling company participate in A-CDM information sharing (regardless of the AOC's remote location).

Towards the end of the boarding process it is discovered that three (3) passengers travelling as a group are missing although they have arrived at the airport and have gone through security.

Missing passenger alert is raised. In order to do this, the tool must be able to trace at least the number of passengers who have passed through check in and security processes (if not actual passenger identities).

The improved granularity provided by A-CDM and TITAN services has enabled the speedy location of the passengers and hence the impact on the operation was minimized.

3.2 Possible Deviation from Baseline Scenario

Shaded boxes in chapter 3.1 correspond to scenario events whose modification could produce a deviation from baseline scenario and consequently smaller verification scenarios in order to highlight the benefits of the tool by demonstrating that additional information is useful to solving problems. According to [2] and [3] the parameters of the baseline scenario that could be modified are the following:

- **Departure/arrival slot**

  The baseline scenario calls for departure slots. By running a scenario that does not require departure slots the user may decide there is more time to find the missing passengers because there is some additional flexibility in the departure time.

- **Weather conditions**

  The baseline scenario calls for fine weather. By running a scenario where bad weather conditions impact the destination airport a degree of urgency to locating the passengers to ensure that the flight will not be affected by the weather can be added.

- **Aircraft fuelling**

  The decision to fuel or not fuel the aircraft may impact the timing of other tasks during the turnaround process. This would be clearly visible in the tool's GUI.

- **Missing passenger status information**

  The baseline scenario defines the missing passengers as high-yield connecting passengers and the airline is willing to locate them and ensure that they are on the flight. By changing their status, the user may change their decision making process. If missing passengers have low status it may be easy for the airline to consider leaving without them.

- **Missing passenger baggage information**

  The baseline scenario says that the missing passengers have checked-in bags. By changing this, they can show that improved knowledge can assist the decision making process. If the missing passengers have no checked baggage, it may be viable to put them on the next flight rather than delaying the flight in order to find them.

- **Missing passenger movements (after check-in)**

  By adding “sightings” of the missing passengers, the user can clearly show the benefits of improved data availability. If the missing passengers have not been seen, they could be almost anywhere in the airport. If they were seen five (5) minutes ago at a particular concessionaire’s business, the area to search is greatly reduced.

3.3 Baseline Scenario Steps and Milestones

The way the scenario affects the intended scope of the tool has been evaluated in detail in [15] and [3], where particular scenario steps are introduced as actual part of the baseline scenario facilitating this way its conduction. A synoptic view of the sequence of the baseline scenario steps is given in Figure 1. The verification, however, focuses on all processes taking place between a/c in-block and off-block time. According to [6] TITAN should complement existing A-CDM milestones between these two turnaround time stamps. TITAN would contribute to further turnaround monitoring (A-CDM extension) through integration of the milestones summarized in Table 1. Shaded cells indicate milestones to be considered in the verification scenario.

Workflow		Milestones (M)			
		Airport with CDM		Complementation through TITAN	
Workflow of TITAN “missing Passenger(s)” Scenario	In-Block	M7	In-Block		
	Handling	M8	Ground Handling starts		
				M17	Close Check-in
				M18	Last Passenger crossing Security Control
				M19	Last Passenger crossing Passport Control
		M9	Final Update of TOBT		
		M10	ATC issues TSAT		
	Boarding	M11	Boarding starts		
	Passenger missing				
	Passenger found			M20	End of De-Boarding
				M21	Last Baggage delivery to hold Baggage Bay
				M22	End of Baggage Unloading
				M23	Close Cargo Doors
				M24	Start of Fuelling
				M25	Remove Pushback
				M26	End of De-Icing
	Ready for Pushback	M12	A/c ready		
		M13	Start-up Request		
	Pushback	M14	Start-up Approval		
	Taxi-out	M15	Off-Block		

Table 1. Complementation of A-CDM Milestones through TITAN [6]

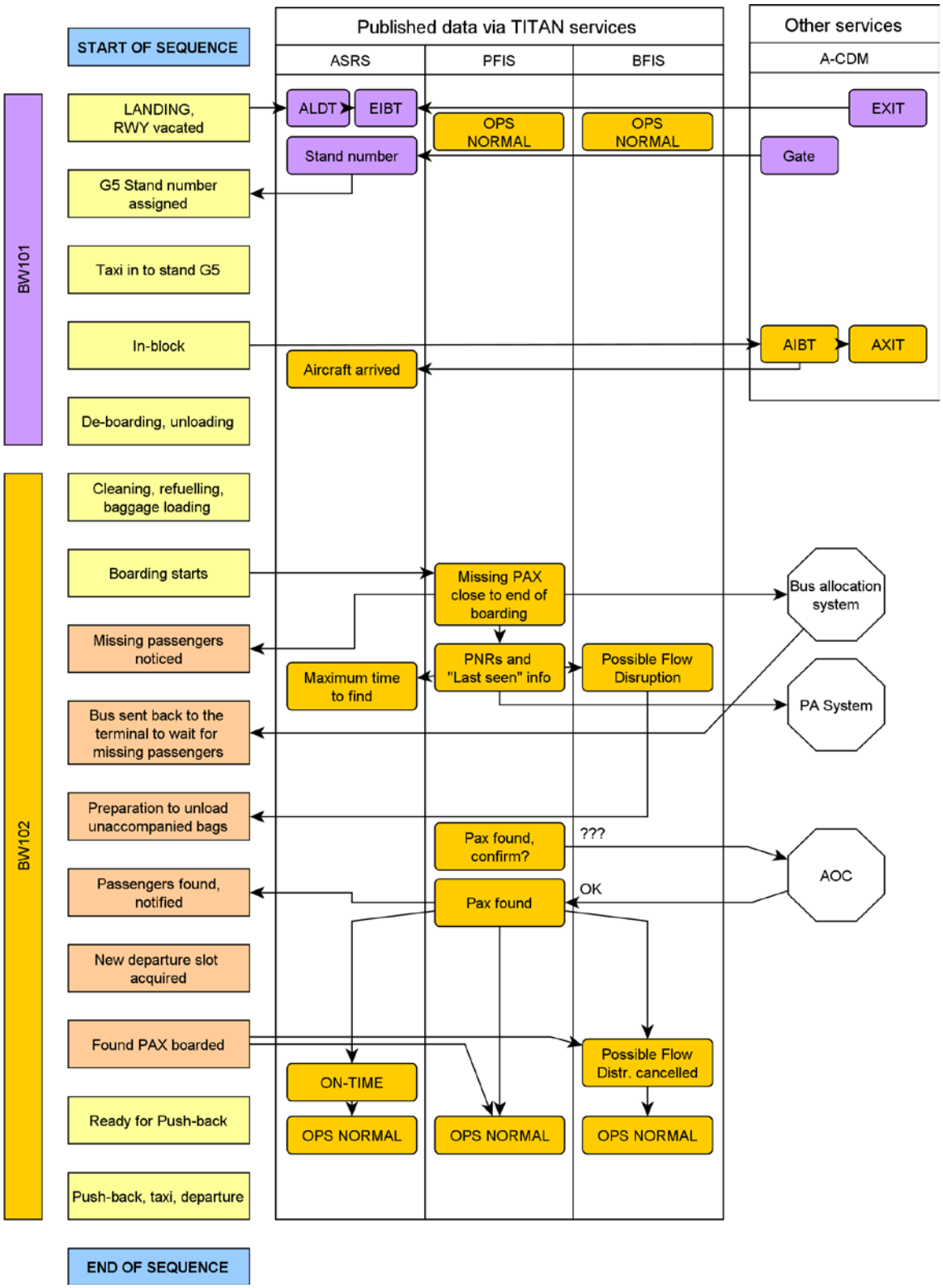

Figure 1. Baseline Scenario Sequence Diagram [13]

4. TOOL REQUIREMENTS TO BE VERIFIED

This chapter is actually the heart of the verification plan document as all functional and non-functional requirements set for TITAN tool in the Turnaround Tool Specification Document [13] are gathered and summarized in requirements verification matrices in an attempt to ease tool verification regarding requirements accomplishment. Requirements verification matrices establish the basis for the verification process as they define i.e. the "shall" statement of each requirement, how each requirement (functional, non-functional) is to be verified or the applicable verification levels. Each requirement listed in the requirements verification matrix is being identified through the following information [14] and [18]:

- **Requirement identification number**

  For each requirement a unique identifier is used.

- **Document/section**

  The document and if possible the section within it that contain the requirement is given.

- **Requirement Description**

  A general high-level (short) description of the requirement is provided.

- **"Shall" statement**

  Text (within reason) of the requirement, i.e., the "shall" text is provided.

- **Verification success criteria**

  The criteria that determine the success/failure of the verification of the requirement are given.

- **Verification method**

  The method according to which the requirement will be verified is provided.

- **Verification Level**

  The product level at which verification occurs is provided (system, service, or end user).

Following information remains the same for all requirements and for that reason it is not included in the requirements verification matrix:

- **Verification Phase**

  Formal tool verification against the specified requirements takes place after tool design and implementation.

- **Performing organization**

  Verification (except for code reviews) will be conducted by RWTH Aachen University.

- **Results**

  Requirements verification results will be consolidated into the final verification report.

Requirements verification (testing) will be conducted in the context of test (use) cases: a logical grouping of functions and performance criteria that is to be tested together. Requirements should be related and easily combined into a reasonable set of test procedure actions. Each test case should contain at least the following information:

- **Test (use) case identification number**

For each test case a unique identifier is used.

- **Test (use) case description**

  A general high-level (short) description of the test (use) case is given providing information about the actor (turnaround stakeholder) and tool functionalities to be verified.

- **Requirement traceability**
- **“Shall” statement**

  Text (within reason) of the test (use) case, i.e. the “shall” text, is provided informing about the main objective of test (use) case.

- **Verification success criteria**

  The criteria that determine the success/failure of the verification of a particular test (use) case expressed in the form of steps to be performed to achieve an expected outcome are given.

4.1 Non-Functional Requirements

In contrast with functional requirements, non-functional requirements are not to be verified through test (use) cases as they do not refer to testable functional tool features. For example there is no meaning in setting a test (use) cases for verifying that the tool is system-oriented built or that HMI is usable and user-customizable. Meeting or not of such non-functional requirements will be the natural outcome of using the tool during the verification process. Exceptions constitute reliability and availability (system downtime testing) that could be also verified in the context of the test (use) cases. However, they do refer to abstract features verified through the whole verification process.

The non-functional requirements that should be met by TITAN software are analytically presented for verification purposes in Table 2.

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
NF-NA-00	Turnaround tool specification document (D4.1) / Section 2.1	Network architecture	The tool shall be built as semi-centralized according to system-oriented architecture principles.	1. TITAN information system (TIS) plays the central role. 2. Services are integrated into the system interacting successfully and being administrated with/by TIS.	Design	System / Service	This is a quality requirement.
NF-PD-00	Turnaround tool specification document (D4.1) / Section 2.1	Platform dependency	Human-machine-interfaces (HMI) shall be specialized to users' needs.	HMI is: 1. adapted to each end user's needs; 2. formed in accordance with each end user's information needs.	Testing / Demonstration	End-user	
NF-R-01	Turnaround tool specification document (D4.1) / Section 2.1	Reliability	Authorized users shall exchange data successfully.	All UIs are capable of: 1. transferring required data; 2. verifying successful data receiving.	Testing	End-user	These are process features.
NF-R-02	Turnaround tool specification document (D4.1) / Section 2.1		End-user interfaces (UI) shall clearly indicate information exchange status.	UI can indicate receiving or pending status of user data sent to the system.	Testing	End-user	
NF-R-03	Turnaround tool specification document (D4.1) / Section 2.1		Data concurrency shall be filtered on the database level.	All system users get exactly the same information as soon as it is made available.	Testing	System / End-user	
NF-R-04	Turnaround tool specification document (D4.1) /		System shall get notified about service unavailability.	TIS, services and users get corresponding (warning) messages for service	Testing	System / Service /	

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
	Section 2.1			unavailability.		End-user	
NF-A-00	Turnaround tool specification document (D4.1) / Section 2.1	Availability	System shall meet minimum operation requirements in case of service downtimes.	Upon the unavailability of a particular service overall system downtime is optimally reduced by implementing specific procedures.	Testing	System	This is a quality requirement. Check if the system can still operate efficiently given a service downtime. This requires "partial system downtime" test to get to a testable behaviour upon system operability.
NF-U-01	Turnaround tool specification document (D4.1) / Section 2.3.4	HMI usability	Information shall be correctly placed on HMI screen.	Information shown in the HMI screen should be: 1. positioned correctly; 2. introduced in the right order; 3. in appropriate grouping.	Testing / Demonstration	End-user	Simply check if the relevant purely technical requirements for the designation of the HMI are met.

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
NF-U-02	Turnaround tool specification document (D4.1) / Section 2.3.4		Adequate colouring shall be implemented.	Under consideration of colour blindness, immediate identification of screen groups and display areas differentiation: 1. Appropriate background colouring; 2. Appropriate screen subsection colouring; 3. Appropriate colour code for particular information status/alarms is used: a) Red - Turnaround is surely delayed (Information Level 3) b) Yellow - Immediate action is required in order to keep the departure on time (Information Level 2) c) Blue - At least one process is delayed, but buffer times ensure on-time operations (Information Level 1) d) Green - Operation normal (Information Level 0)	Testing / Demonstration	End-user	
NF-U-03	Turnaround tool specification document (D4.1) /		Appropriate fonts shall be implemented.	The implemented fonts are: 1. common; 2. of appropriate size to enable	Testing / Demonstration	End-user	

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
	Section 2.3.4			easy key information reading.			
NF-U-04	Turnaround tool specification document (D4.1) / Section 2.3.4		Appropriate unit systems shall be implemented.	The implemented unit systems are uniform.	Testing / Demonstration	End-user	

Table 2. TITAN Tool non-functional Requirements Verification Matrix

4.2 Functional Requirements

Also indicated in the Turnaround Tool Specification Document [13], functional requirements can be divided into two main categories depending on the respective reference system:

- **General (system) functional requirements**

  This requirements cluster refers to the basic/principal functions of the system and its components, i.e. monitoring the system, accessing the system, exchanging information, logging out of the system, searching into the system etc.

- **Specific (UI) functional requirements**

  This requirements cluster refers to functions of the interface between the system and the end-user, i.e. layout, customizability, information content etc.

  UI functional requirements can be further divided into particular categories:

  - Common requirements for all UIs;
  - Requirements for summarized information presentation purposes;
  - Requirements for more detailed information presentation purposes;
  - Requirements for process-detailed information presentation purposes.

However, all these requirements will be presented in a uniform way. Two different requirements verification matrices will be though used for distinction reasons; one for general (system) requirements and one for specific (UI) requirements.

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
F-S-AD-01	Turnaround tool specification document (D4.1) / Section 2.4	Administration[1]	Administrator shall manage users and their sessions.	Through a simple client administrator adds/deletes users and edits their profiles and roles. Administrator manages user sessions by checking credentials and issuing unique session IDs. After initial authentication, session IDs are used for maintaining message authenticity, until session timeout.	Testing	System (Admin)	
F-S-AD-02	Turnaround tool specification document (D4.1) / Section 2.4		Administrator shall manage services and their sessions.	Through a simple client administrator starts up/shuts down services and manages their sessions.	Testing	System (Admin)	
F-S-SAU-01	Turnaround tool specification document (D4.1) / Section 2.4	System access and use	Users shall log in/out of TIS successfully (from the user's point of view).	Users log in TIS that subscribes them to services based on their roles.	Testing	End-user	
F-S-SAU-02	Turnaround tool specification document (D4.1) / Section 2.4		Services shall subscribe to TIS for getting necessary information.	Services tell TIS/administrator that they need to get notified about specific event types they subscribe for (continuously or temporarily).	Testing	Service	
F-S-EH-01	Turnaround tool specification document (D4.1) / Section 2.4	Event handling (submission and distribution)	Events triggered by end users shall be processed via services to TIS (first submission	End users submit newly available information regarding events (incl. milestones and warnings) they are responsible for to services.	Testing	End-user	

[1] Verification of the administration requirements depends on the level of detail of the administration interface to be finally implemented by the development team.

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
			or updating).				
F-S-EH-02	Turnaround tool specification document (D4.1) / Section 2.4		TIS notifies subscribed services and users accessing them about new/updated events.	On the basis of the subscribers' list notifications are sent to services subscribed for specific event types by TIS.	Testing	System (TIS)	
F-S-IC-00	Turnaround tool specification document (D4.1) / Section 2.3.4	Information classification[2]	TIS classifies information to be distributed to services and users.	The system assesses the levels of information upon the data available.	Testing	System (TIS)	
F-S-ASRF-00	Turnaround tool specification document (D4.1) / Section 2.4	A/c status reporting / forecasting	Upon users' requests ASRS shall report and forecast a/c status back to them.	ASRS gathers a/c status relevant information from all users involved in its turnaround and uses it for: 1. reporting the actual status and 2. forecasting a near-future status of a specified a/c including milestone information. Upon stepping a Milestone forward, ASRS broadcasts a notification event to the TIS.	Testing	Service (ASRS)	

[2] Verification of the information classification requirement depends on the level of detail of the information classification logic to be finally implemented by the development team. This requirement is indirectly considered in the formulation of requirements F-UI-D-02/03 of Table 4.

Requirement ID-Number	Document / Section	Requirement Description	"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
F-S-PBS-01	Turnaround tool specification document (D4.1) / Section 2.4	Passenger / Baggage search	ASRS shall provide deadline information for finding missing passenger regarding on-time performance.	ASRS combines TIS data with output data of a/c status forecaster to provide so called "maximum time to find" type information when a passenger is missing.	Testing	Service (ASRS)	
F-S-PBS-02	Turnaround tool specification document (D4.1) / Section 2.4		PFIS shall search for last trace of a missing passenger (i.e. by concessionaires) in order to give a safe prediction about his position.	PFIS mines data in the TIS log to: 1. find "last seen" type records related to specified passengers; 2. guess their location within the terminal building.	Testing	Service (PFIS)	
F-S-PBS-03	Turnaround tool specification document (D4.1) / Section 2.4		PFIS shall generate a PNR formatted output for a missing passenger.	Upon request PFIS provides PNR-s of specified passengers.	Testing	Service (PFIS)	
F-S-PBS-04	Turnaround tool specification document (D4.1) / Section 2.4		BFIS shall search for last trace of a missing passenger's baggage (i.e. by baggage screening station) in order to give a safe prediction about its position.	BFIS mines data in the TIS log to: 1. find "last seen" type records related to specified bag-tags; 2. guess their location within the airport or name the containing pallet.	Testing	Service (BFIS)	

Table 3. TITAN Tool general functional Requirements Verification Matrix

Requirement ID-Number	Document / Section	Requirement Description		"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
F-UI-C-01	Turnaround tool specification document (D4.1) / Section 2.3.3	UI common	Login /Logout information	User information (name, profile) shall be visible after successful login and logout shall be easily executable.	After successful login, user name is displayed in a consistent UI location and user profile pops up i.e. by clicking on it. Logout/confirmation (i.e. pop up window) are possible through a logout button/link.	Testing / Demonstration	End-user	
F-UI-C-02	Turnaround tool specification document (D4.1) / Section 2.3.3		Federated login	Login shall be centralized to a login screen.	Upon first http request to any TITAN domains, the login screen pops up, and then the user is redirected to the requested page.	Testing / Demonstration	End-user	
F-UI-C-03	Turnaround tool specification document (D4.1) / Section 2.3.3		Time display	Time shall be visible to all users.	Common local time is displayed permanently on all UIs.	Testing / Demonstration	End-user	
F-UI-C-04	Turnaround tool specification document (D4.1) / Section 2.3.3		Touch optimization	The interacting UI shall be appropriately sized.	The views are size-optimized to enable convenient use of both touch-enabled and conventional displays on desktops and tablets (i.e. iPads).	Testing / Demonstration	End-user	
F-UI-C-05	Turnaround tool specification document (D4.1) / Section 2.3.3		Data refreshment / update	Data on the current UI view shall be automatically refreshed as soon as their updates shall be made available.	The web UI is designed to enable refreshment of individual page portions containing updated data without needing to reload the whole page at once.	Testing / Demonstration	End-user	
F-UI-S-01	Turnaround tool specification document (D4.1) / Section 2.3.3	[illegible] (default) screen	Layout	Specific (common) flight information shall be fixed presented on the UI default screen view.	Common flight information such as status, number, departure/destination airport and EIBT/EOBT is fixed, while other such as a/c type or number of passengers can be replaced.	Testing / Demonstration	End-user	

Requirement ID-Number	Document / Section	Requirement Description		"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
F-UI-S-02	Turnaround tool specification document (D4.1) / Section 2.3.3		Warnings	Flight status warnings shall be related to a specific colour code.	Colour coding (NF-U-02) is used in close relation to the fixed columns (i.e. via a frame) and the replaceable columns as well	Testing / Demonstration	End-user	
F-UI-S-03	Turnaround tool specification document (D4.1) / Section 2.3.3		Customizability	The UI default screen view shall be user-customizable regarding non-fixed flight information enabling also save option of customized layout.	Unnecessary flight information columns can be replaced by the desired ones selected from a respective list* (that i.e. pops up when clicking on a columns header). Server registers the layout changes for the particular user, who gets his last saved UI layout when logging in from a different client. *i.e. a/c type, tail number, passenger number, C/A/ETOT, E/ALDT, AI/OBT, status, requested services etc.	Testing / Demonstration	End-user	
F-UI-S-04	Turnaround tool specification document (D4.1) / Section 2.3.3		Navigation	Further flight specific information shall be accessed through the UI default screen view.	The UI detail screen view of a particular flight pair comes up through the UI default screen view (i.e. by clicking on fixed flight information columns).	Testing / Demonstration	End-user	
F-UI-D-01	Turnaround tool specification document (D4.1) / Section 2.3.3	UI detail screen view	In line with UI summary (default) screen view requirements	UI summary screen view requirements regarding layout, warnings, customizability and navigation shall apply to the detail view too.	Fixed flight information is displayed similarly to the summary view. Colour coding similar to that of the summary view is used on tabs bar and Gantt views of different tab panels. Unnecessary flight data (of a customisable list) can be replaced by the desired ones and customized layout can be saved and retrieved just like in the summary view. Summary view can be retrieved with the closed flight still in focus (i.e. by clicking on	Testing / Demonstration	End-user	

Requirement ID-Number	Document / Section	Requirement Description		"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
					the fixed flight data or a "back" button).			
F-UI-D-02	Turnaround tool specification document (D4.1) / Section 2.3.3		Active key information on a/c turnaround processes	The actual progress of a/c turnaround processes related to milestones shall be displayed through tabs containing active (real time) textual key content in line with the applied colour coding.	In line with the applied colour coding, active textual key content provided through a clickable tabs bar informs about the level of completion of particular a/c turnaround processes related to critical milestones. Following tabs are available: general flight information, check-in, passenger security control, passenger (de)boarding, baggage (un)loading, other services (cleaning/catering), fuelling, start-up and de-icing. Following common messages is displayed: 1. Operations normal (Green – information level 0) 2. Not applicable (Gray) 3. Short status message introduced by the user 4. The highest information level or "n ISSUES" text, where "n" is the number of information of similar level.	Testing / Demonstration	End-user	
F-UI-D-03	Turnaround tool specification document (D4.1) / Section 2.3.3		Active detailed information on a/c turnaround processes	All information capable of giving a detailed view over the conduction of a/c turnaround processes related to milestones	In line with the applied colour coding, an active process detail section can be accessed by selecting one of the clickable tabs for the processes listed in F-UID-02 informing about the conduction of particular a/c turnaround processes and granting	Testing / Demonstration	End-user	

Requirement ID-Number	Document / Section	Requirement Description		"Shall" Statement	Verification Success Criteria	Verification Method	Verification Level	Notes
				shall be displayed in an active (real time) process detail section in line with the applied colour coding and the information overload prevention.	particular users write access over particular information. The tabs listed in F-UID-02 apply also here. In compliance with information overload prevention, the process detail section contains following minimum information clusters[3]: 1. Process start/end time and milestone information; 2. Gantt diagram of affected flights/processes in case of expected delay in the current process; 3. Stand/gate number information; 4. Process-specific information.			

Table 4. TITAN Tool specific UI functional Requirements Verification Matrix

[3] For further details on minimum content of active detailed information of each tab please see Appendix A!

4.3 Requirements Summary

All functional and non-functional requirements are summarized for traceability reasons in Table 5.

Requirement ID-Number	Requirement Description	
NF-NA-00	Network architecture	
NF-PD-00	Platform dependency	
NF-R-01	Reliability	Successful data exchange
NF-R-02		Data exchange status
NF-R-03		Data concurrency
NF-R-04		Service unavailability notification
NF-A-00	Availability	
NF-U-01	HMI usability	Information screen positioning
NF-U-02		Colouring
NF-U-03		Fonts
NF-U-04		Unit systems
F-S-AD-01	Administration	User (session) administration
F-S-AD-02		Service (session) administration
F-S-SAU-01	System access and use	User log-in and log-out
F-S-SAU-02		Service subscription
F-S-EH-01	Event handling	Event submission
F-S-EH-02		Event distribution
F-S-IC-00	Information classification	
F-S-ASRF-00	A/c status reporting / forecasting	
F-S-PBS-01	Passenger / Baggage search	Deadline information
F-S-PBS-02		Passenger search
F-S-PBS-03		PNR
F-S-PBS-04		Baggage search
F-UI-C-01	UI common	Login / Logout information
F-UI-C-02		Federated login
F-UI-C-03		Time display
F-UI-C-04		Touch optimization
F-UI-C-05		Data refreshment / update
F-UI-S-01	UI summary (default) screen view	Layout
F-UI-S-02		Warnings
F-UI-S-03		Customizability
F-UI-S-04		Navigation
F-UI-D-01	UI detail screen view	In line with UI summary (default) screen view requirements
F-UI-D-02		Active key information on a/c turnaround processes
F-UI-D-03		Active detailed information on a/c turnaround processes

Table 5. Requirements Summary

4.4 High-Level Use-Cases

The test (use) cases of the TITAN tool verification plan have to be defined in absolute accordance with:

- the verification scenario described briefly in chapter 3 and in detail in [15];
- the actors (turnaround stakeholders) and functionalities defined for implementation by the TITAN tool in the Turnaround Tool Specification Document [13].

The outcome of the test (use) cases should verify that all assumptions made in the verification scenario are correct and the main goal of the tool can be achieved; whether the missing passenger is found and boarded or not, the TITAN tool assists/supports all involved actors in decision-making by making them aware of the situation and its impact on the particular flight or any other affected a/c turnaround process and enabling them to minimize or even eliminate it. In other words it should be proved through the test (use) cases that the tool is able to present appropriate data to different types of user in order to enable the scenario's goal to be met.

Firstly, an efficient filtering of all information necessary for the successful running of the verification scenario is required for an accurate definition of the test (use) cases. For that reason the particular scenario steps were examined in detail and a list of all information required to be exchanged during the scenario was produced and is presented in Table 6.

For the successful completion of the process steps of the verification scenario (resulting in the accomplishment of the scenario goal) specific information is necessary to be published and provided by specific actors (turnaround stakeholders) through specific tool services making them available to anyone who actually needs it. Particular information constitutes A-CDM milestones (red coloured cells) and TITAN additional milestones (green coloured cells) [6]. Further information should be also considered as milestones in the particular verification scenario (yellow coloured cells). Most important scenario-related information as well as its provider together with any services related to its distribution into the information sharing system will be further considered for the definition of the test (use) cases of the verification scenario (bold-framed cells).

Type of Information	Information Provider	Related Services
Actual Landing Time of a/c	ACDM	ASRS, ACDM
Actual Taxi-in Time of a/c	ACDM	ASRS, ACDM
Estimated/Actual In-Block Time of a/c	AOC (Flight Crew), **GH**	ASRS, ACDM
Park Position (Stand) and Gate finally allocated to a/c	AOP	ASRS, AIRS, PFIS
Service Normal Operation Notification	PFIS, BFIS (internal check)	PFIS, BFIS
Passenger Check-in (Open/Close Time)	AOC (Check-In Desk)	BFIS, AIRS
Number of Checked-In Passenger (inc. On-line)	AOC (Check-In Desk)	PFIS, AIRS
Number of Passenger checked-in Baggage	AOC (Check-In Desk)	BFIS, PFIS, AIRS
Passenger Group Travelling Information	AOC (Check-In Desk)	PFIS, AIRS
Passenger connecting flight Information	AOC (Check-In Desk)	PFIS, AIRS
Passenger Security Control (First/Last passenger of the corresponding flight)	AOP (Passenger Security Control Personnel)	PFIS
Baggage Drop-off Information	AOC (Baggage Drop-off Station)	BFIS, PFIS, AIRS
Baggage Security-Screening (First/Last baggage of the corresponding flight)	GH (Baggage Agent) or AOP	BFIS, AIRS
Baggage Sorting (First/Last baggage of the corresponding flight)	GH (Baggage Agent) or AOP	BFIS, AIRS
Ground Handling (Start/End Time) [4]	GH	AIRS, ASRS
Passenger De-Boarding (Start/End Time)	AOC (Flight Crew), GH	PFIS, AIRS, ASRS
Baggage Unloading (Start/End Time)	Equipment Operator (Baggage Handler)	BFIS, ASRS
A/c Cleaning/Catering (Start/End Time)	GH	ASRS
A/c Fuelling (Start/End Time)	GH	AIRS, ASRS
Passenger Boarding (Start/End Time)	AOC (Gate, Flight Crew)	PFIS, AIRS, ASRS
Baggage Loading (Start/End Time)	Equipment Operator (Baggage Handler)	BFIS, ASRS
Ground Transportation Resources Status (Busses for Passenger Boarding)	Equipment Operator (Bus)	AIRS
Missing Passenger Notification	AOC (Gate)	PFIS
Missing Passenger Class Notification	AOC (Gate)	PFIS
Order to wait or not for missing Passenger (according to its Class)	AOC	PFIS

[4] This information is automatically produced as soon as first/last handling sub-activity starts/ends

Type of Information	Information Provider	Related Services
Order for Bus to go and wait for missing Passenger	Equipment Operator (Bus)	PFIS, AIRS
Order for missing Passenger Baggage unloading	GH	BFIS, AIRS
Passenger Name Record (PNR) Publication	PFIS (internal artefact)	PFIS
PNR Information Submission (last seen type Information, missing Passenger Movement)	AOP (Concessionaires)	PFIS
Missing Passenger Baggage Detection	BFIS (internal check)	BFIS
Possible Flow Disruption Notification	BFIS (internal artefact)	BFIS
Public Annunciator System Triggering	PFIS (internal artefact)	PFIS
Maximum Time to find missing Passenger	ASRS (internal artefact)	ASRS
A/c Status regarding Delay due to Search for missing Passenger	ASRS (internal artefact)	ASRS
Status of connecting Flight	ACDM, AOC	ACDM
Slot Status of corresponding Flight	ACDM, ASRS	ASRS, ACDM
Estimated In-Block Time of next Flight to which currently occupied Gate is allocated	ACDM, AOC (other)	ACDM, ASRS
Container of missing Passenger Baggage Information	BFIS (internal artefact)	BFIS
Time for Removal of missing Passenger Baggage Container (+ latest completion)	Equipment Operator (Baggage Handler)	BFIS
Missing Passenger found Notification (AOC confirmation necessary)	PFIS (internal artefact)	PFIS
Cancelation of missing Passenger baggage status changes	BFIS (internal artefact)	BFIS
A/c on Time Status Notification (within buffer time)	ASRS (internal artefact)	ASRS
A/c ready for Pushback	AOC (Flight Crew), **GH**	ASRS
Start-Up Request	AOC (Flight Crew)	ASRS
Start-Up Approval - Actual Start-Up Time (matches the Target Time?)	ATC - ACDM	ASRS, ACDM
Actual Off-Block Time (Pushback)	GH	ASRS, AIRS
Actual Taxi-Out Time	ACDM	ASRS, ACDM
Actual Take-Off Time (matches final Update of Target Time?)	ACDM	ASRS, ACDM

Table 6. Filtering of Information Exchange necessary for the Verification Scenario

For each one of the pre-selected actors we have to check if corresponding (pre-selected) TITAN Tool functionalities of services that each actor would use in the missing passenger scenario for providing/getting any of the previously defined information he needs can be successfully used. According to [13] following actors and functionalities summarized in Table 7 should be considered for the definition of the test (use) cases.

Actor			**Functionalities (Services)**					
			User/service administration	Status reporting	Self allocation and task schedule (AIRS)	PAX info and "last seen" log (PFIS)	Baggage info and "last seen" log (PFIS)	Type and configuration info (ASRS)
Airport Operator	Administrator		X	X				
	Passenger Security Control Personnel			X	X	X		
	Concessionaires[5]			X	X	X		
A/c Operator	Flight Crew[6]			X				X
	Check-in Agent	Check-in Desk		X	X	X	X	
		Gate		X	X	X (info)		
Ground Handler	Baggage Agent			X	X		X	
	Equipment Operator (incl. bus)			X	X	X (info)		
	Baggage Handler			X	X		X	

Table 7. Actors and Functionalities for the Test (Use) Cases [13]

The input of the test (use) case is defined by identifying the actor (Table 7), the service functionalities (Table 7) and the related requirements as well as by the verification scenario and the information exchange needed for its successful completion (Table 6). The steps to be followed in the test (use) case are described in the verification success criteria that represent the test (use) case outcome to be expected. The functionalities set in [13] for the different actors to be

[5] Previously also referred to as Duty Free Shops in the 1st issue of the Operational Concept

[6] Previously also referred to as Cockpit Crew in the 1st issue of the Operational Concept

implemented by the tool serve as the basis that is to be reinforced by the information exchange needs of the particular verification scenario.

All test (use) cases that will be part of the TITAN tool verification are summarized in the test (use) case verification matrix of Table 8 on the basis of the above principles. A traceability matrix to correlate test (use) cases with requirements to be tested is integrated into this table.

In order to prevent information duplication, following scenario steps apply to all test (use) cases summarized in Table 8 by the time missing passenger alert is raised:

Gate (agent) submits missing passenger notification via PFIS as soon as missing passenger is detected submitting also missing passenger class information (economy, business).

PFIS raises alarm due to missing passenger (subsequently publishing PNR) and BFIS notifies about possible flow disruption due to missing passenger's baggage detection (subsequently publishing information of missing passenger's baggage container ID).

AOC submits its decision whether to wait or not for missing passenger according to his class via PFIS.

All searching activities begin via PFIS and BFIS. Last trace of missing passenger and his baggage (last seen log) is retrieved by mining TIS log and an estimation of their position in terminal can be made (activating subsequently public annunciation system):

- If check-in desk submission of PNR information via PFIS is the last passenger trace information, passenger can be found somewhere between check-in and security control.
- If passenger security control submission of PNR information via PFIS is the last passenger trace information, passenger can be found somewhere between security control and gate.
- If baggage is already processed to baggage security screening, sorting and loading systems and baggage agent's submission of baggage container ID information via BFIS is the last missing passenger's baggage trace information, missing passenger's baggage is not yet loaded into the aircraft.
- If baggage handler submission of baggage container ID information via BFIS is the last missing passenger's baggage trace information, it is confirmed that baggage is already loaded into the a/c and operation controller submits (after AOC confirmation to wait for missing passenger) order to unload missing passenger's baggage (until he is found).

During such activities ASRS combines actual (real-time) data fed into TIS by services and stakeholders effectively taking part in the searching process enabling this way ASRF to calculate and notify about maximum time to find missing passenger (let him board and load again his baggage) on the basis of initial departure planning and available buffer time. In parallel, it publishes actual a/c status regarding delay (exceeding initial planning plus buffer time) due to missing passenger search. On the meanwhile, stand "occupied" is continuously updated.

Equipment operator submits bus order to come back and wait for missing passenger (and so occupied status of its resource) via PFIS and AIRS. Bus driver is notified via PFIS and AIRS that he needs to go back to gate and wait until missing passenger is found in order to transport him to the a/c in order to minimize possible flight delay.

As soon as missing passenger is found:

- PFIS publishes "passenger found" notification (after necessary AOC confirmation depending on passenger class).
- BFIS cancels status changes of missing passenger's baggage.

- Operation controller (ground handling) submits order of loading again missing passenger's baggage into the a/c.
- ASRS notifies about on time status of a/c (return to normal operations).
- Flight crew can submit information about E/AOBT of ready for pushback a/c.
- Missing passenger proceeds to the bus and gate (check-in agent) submits missing passenger boarding information via PFIS.
- Bus driver submits "missing passenger's successful transportation to the a/c" notification via PFIS and his new unoccupied status via AIRS.

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
T(U)C-AD	Administrator	User administration (TIS) Service administration (TIS) Status reporting (TIS, AIRS)	F-S-AD-01/02 (user and service administration) F-S-SAU-01/02 (user log in/out and service subscription) F-S-EH-02 (event distribution)	Administrator shall manage all logged users and subscribed services as well as their sessions, and shall submit status information.	Administrator adds/starts up and deletes/shuts down users/services (approves log in/out and subscription), edits their profiles and manages their sessions via TIS. Administrator sends notifications to subscribed services via TIS. Administrator submits status information via TIS as a means of supervising system normal operation.	
T(U)C-PSCP	Passenger Security Control Personnel	Self allocation and task schedule (AIRS) Passenger information and "last seen" log (PFIS) Status reporting (PFIS, AIRS)	F-S-SAU-01/02 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-PBS-02/03 (Passenger search and PNR) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	Passenger Security Control Personnel shall get/give information about (planned) allocation/schedule of its resources, shall submit status information and shall effectively take part in the process of finding missing passenger.	Passenger Security Control Personnel gets and gives information via AIRS about its (planned/approved) task schedule and allocation of its resource. Passenger Security Control Personnel submits status information via AIRS and PFIS (open 24h of 7 days of week). Passenger Security Control Personnel submits real-time information about number of screened passengers (last screened passenger of the corresponding flight) via PFIS and AIRS as well as PNR-formatted information for each screened passenger via PFIS (last seen log).	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
T(U)C-CS	Concessionaires	Self allocation and task schedule (AIRS) Passenger information and "last seen" log (PFIS) Status reporting (AIRS, PFIS)	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-PBS-02/03 (Passenger search and PNR) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	Concessionaires shall get/give information about (planned) allocation/schedule of its resources, shall submit status information and shall effectively take part in the process of finding missing passenger.	Concessionaires get and give information via AIRS about his (planned/approved) task schedule and the allocation of its resource. Concessionaires submit status information via AIRS and PFIS (shop open). Concessionaires submit real-time information about number of passengers that visited it (last passenger of the corresponding flight) via PFIS and AIRS as well as PNR-formatted information for each visiting passenger via PFIS (last seen log).	
T(U)C-FC	Flight Crew	Type and configuration information (ASRS) Status reporting (PFIS, AIRS, ASRS)	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-ASRF-00	Flight Crew shall get/give information about a/c type and configuration as well as about completion of particular a/c handling milestones that shall enable a/c future status forecasting and deadline calculation for missing passenger searching and	Flight Crew gets and gives information about a/c type and configuration via ASRS. Flight Crew submits status information via ASRS, PFIS and AIRS (E/AIBT of a/c, assigned stand/gate, end time of passenger de-boarding, start/processing/end time of passenger boarding, number of passengers on board, AOBT of a/c, a/c ready for pushback, R/ASAT, remove pushback).	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
			(A/c status report, future status forecast) F-S-PBS-01 (Deadline functionality) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	shall submit status information.		
T(U)C-CA I	Check-in Agent - Check-In Desk	Self allocation and task schedule (AIRS) Passenger information and "last seen" log (PFIS) Baggage information and "last seen" log (BFIS) Status reporting (PFIS, BFIS, AIRS)	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-PBS-02/03/04 (Passenger/Baggage search and PNR) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements)	Check-In Desk shall get/give information about (planned) allocation/schedule of its resources, shall submit status information and shall effectively take part in the process of finding missing passenger and his baggage.	Check-In Desk gets and gives information via AIRS about its (planned/approved) task schedule and allocation of its resources. Check-In Desk submits status information via AIRS and BFIS (check-in open/processing/close time). Check-In Desk submits real-time information about number of not yet checked-in passengers and PNR-formatted and baggage-container-ID-formatted information for each checked-in passenger and baggage via PFIS, BFIS and AIRS (last seen log) in form of PNL (incl. number of checked-in baggage per passenger).	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
			F-UI-D-01/02/03 (UI detailed screen view requirements)			
T(U)C-CA II	Check-in Agent - Gate Allocator	Self allocation and task schedule (AIRS) Passenger information (PFIS) Status reporting (PFIS, AIRS, ASRS)	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-ASRF-00 (A/c status report, future status forecast) F-S-PBS-01 (Deadline functionality) F-S-PBS-03 (PNR functionality) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	Gate shall get/give information about (planned) allocation/schedule of its resources, shall submit status information and shall notify about missing passenger.	Gate gets and gives information via AIRS about its (planned/approved) task schedule and allocation of its resources. Gate submits status information via AIRS, PFIS and ASRS (assigned gate, gate open/close time, boarding start/processing/end time). Gate submits real-time information about number of boarded passengers (last boarded passenger of the corresponding flight on or not on time). Gate submits PNR-formatted information for each boarded passenger and a list of not-yet-boarded passengers via PFIS. Gate submits missing passenger notification accompanied by missing passenger class information via PFIS.	
T(U)C-EO I	Equipment	Self allocation and task schedule	F-S-SAU-00	Equipment Operator shall get/give information about	Equipment Operator gets and gives information via AIRS about its	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
	Operator	(AIRS) Status reporting (PFIS, AIRS, ASRS)	(user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-ASRF-00 (A/c status report, future status forecast) F-S-PBS-01 (Deadline functionality) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	(planned) allocation/schedule of its resources, shall submit status information and shall effectively take part in the process of transporting missing passenger to a/c (bus).	(planned/approved) task schedule and allocation of its resources. Equipment Operator submits status information via PFIS, AIRS and ASRS (assigned stand/gate, de-boarding start time, bus departure/arrival time at /from gate/a/c, bus transportation time, bus status, notification about successful transportation of missing passenger, remove pushback). Equipment operator submits bus order to come back and wait for missing passenger via PFIS and AIRS.	
T(U)C-EO II	Equipment Operator - Baggage Handler	Self allocation and task schedule (AIRS) Baggage information and "last seen" log (BFIS) Status reporting	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification)	Baggage Handler shall get/give information about (planned) allocation/schedule of its resources, shall submit status information and shall effectively take part in the process of finding and preparing to unload	Baggage Handler gets and gives information via AIRS about its (planned/approved) task schedule and allocation of its resources. Baggage Handler submits status information via BFIS and ASRS (assigned stand, start/processing/end time of baggage (un)loading) Baggage Handler submits information about	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
		(BFIS, AIRS, ASRS)	F-S-ASRF-00 (A/c status report, future status forecast) F-S-PBS-01 (Deadline functionality) F-S-PBS-04 (Baggage search functionality) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)	missing passenger's baggage.	maximum removal time of container of missing passenger's baggage, latest possible completion time and actual completion time via BFIS after getting order from operation controller to unload missing passenger's baggage. Baggage Handler submits baggage-container-ID-formatted information via BFIS (last seen log). Beyond the verification scenario Baggage Handler submits additional status information via BFIS and ASRS (loading instruction, loading problem, last bag delivered to hold baggage bay, baggage cannot be delivered/loaded, end of load baggage, close cargo doors, resource malfunction).	
T(U)C-BA	Baggage Agent	Self allocation and task schedule (AIRS) Baggage information and "last seen" log (BFIS) Status reporting (BFIS, AIRS, ASRS)	F-S-SAU-00 (user log in/out and service subscription) F-S-EH-01/02 (event submission and distribution) F-S-IC-00 (information classification) F-S-ASRF-00 (A/c status report, future status forecast) F-S-PBS-01	Baggage Agent shall get/give information about (planned) allocation/schedule of its resources, shall submits status information and shall effectively take part in the process of finding and preparing to unload missing passenger's baggage.	Baggage agent gets and gives information via AIRS about its (planned/approved) task schedule and allocation of its resources. Baggage Agent submits status information via BFIS and ASRS (assigned stand, start/processing/end time of baggage sorting and loading into containers/carriages and transportation time to a/c). Baggage Agent submits baggage-container-ID-formatted information via BFIS (last seen log).	

Test (Use) Case ID-Number	Test (Use) Case Description		Requirement Traceability	"Shall" Statement	Verification Success Criteria	Notes
	Turnaround Stakeholder (Actor)	Functionality (Service)				
			(Deadline functionality) F-S-PBS-04 (Baggage search functionality) F-UI-C-01/02/03/04/05 (UI common requirements) F-UI-S-01/02/03/04/05 (UI default screen view requirements) F-UI-D-01/02/03 (UI detailed screen view requirements)			
T(U)C-SDT	Non specified	Non specified	NF-R-04 (Service unavailability notification) NF-A-00 (Availability)	Upon a particular service downtime, turnaround stakeholders as well as services shall get notified about service unavailability and system shall meet minimum operation requirements.	TIS, services and end-users get warning message for particular service unavailability and overall system downtime is optimally reduced. The procedure of searching for missing passenger(s) can be resumed and the main scenario goal is achieved; whether the missing passenger is found and boarded or not, the TITAN tool assists/supports all involved actors in decision-making by making them aware of the situation and its impact on the particular flight or any other affected a/c turnaround process and enabling them to minimize or even eliminate it. (Procedures described previously apply here too.)	

Table 8. Test (Use) Cases Verification Matrix

5. VERIFICATION CONDUCTION DETAILS

The responsibility as well as the time frame of the verification conduction is given in Table 9. The agile verification method to be implemented is not supposed to substitute the final verification, whose results will be consolidated in Turnaround Tool Verification Document (D4.5).

		Performing Organisation	Phase	Notes
Verification Type	Reviews or inspections of the software code	Development team	During development and implementation phase	Not part of formal tool verification
	Verification of software executables ("smoke testing")	Verification team	During development and implementation phase	
	Formal tool verification against requirements	Verification team	After development and implementation phase	

Table 9. Verification Conduction Responsibility and Time Frame

Focusing on formal tool verification, the demonstrator accompanied by a short use manual will be delivered to the verification team) enabling its verification against requirements through defined test (use) cases.

Testing at once (simultaneously) all the actors specified in the test cases - what would correspond to a realistic representation of a single turnaround process - would require a formal verification procedure, where different people would play different roles. In such a case the preparation and conduction time costs would be quite high. However, given that the purpose of the tool is basically situational awareness and that the particular verification scenario includes few actions, such a verification approach would not be of high benefit, as the investment in time and effort would not add anything valuable. On the other hand, testing the tool by implementing the different actors and the different test (use) cases in different software runs could be performed by the verification team making the above procedure unnecessary. This is further facilitated by the following two facts:

- Each test (use) case corresponds to a single actor (turnaround stakeholder).
- Within the demonstrator, the individual actors just show different data. When running the test as one actor, the actions of other actors can be emulated. So called "event players" just emit events according to a predefined schedule embedded in the system (emulation of live data delivery just as supposed to be done with CDM).

The verification procedure will be kept as simple as possible. The specification document delivered the list of requirements enabling the structuring of the current document (verification plan) that sets requirements verification in the context of the defined test (use) cases.

ANNEX I – PROCESS INFORMATION ON UI DETAIL VIEW

For each one of the tabs of requirement F-UI-D-03 (Table 4) the active textual detailed content available for each a/c turnaround process may include:

- General flight information tab
  - Estimated/actual arrival (L+IB)/departure (OB+TO) times (information level 0,1, 2, 3)
  - Assigned stand/gate (information level 0)
  - Textual indication of all general sub-process information on level other than 0
  - Gantt of affected flights or processes, if delay is expected (information level 1, 2, 3)
- Check-in tab
  - Check-in open/close time (information level 0)
  - Number of not checked-in passengers (information level 0)
  - PNL (information level 0)
- Passenger security control tab
  - Last passenger crossing security control (information level 0)
- Baggage unloading tab
  - Baggage unloading start/end time (information level 0)
  - Allocated stand (information level 0)
- Passenger de-boarding tab
  - Passenger de-boarding start/end time (information level 0)
  - Allocated gate (information level 0)
- Other service tab (out of scope for the verification scenario)
- Fuelling tab (out of scope for the verification scenario)
- Baggage loading tab
  - Stand number (information level 0)
  - Gantt of affected flights or processes, if delay is expected (information level 1, 2, 3)
  - Loading instruction (information level 0)
  - Baggage loading start/end time (information level 0)
  - Any problem when loading (information level 1, 2, 3)
  - Last bag delivered to hold baggage bay (information level 0)
  - Baggage cannot be delivered (information level 1, 2, 3)
  - End of load baggage (information level 0)
  - Baggage cannot be loaded (information level 1, 2, 3)
  - Close cargo doors (information level 0)
  - Malfunction of any resource (information level 1, 2, 3)

  - Order to unload missing passengers' baggage (information level 1, 2, 3)
- Passenger boarding tab
  - Gate open/close time (information level 0, 1, 2, 3)
  - Gate number (information level 0)
  - EOBT (information level 0)
  - Passenger boarding start/end time (information level 0)
  - Last passenger at gate (information level 0)
  - Last passenger at gate not on time (information level 1, 2, 3)
  - Number of passenger on board (information level 0)
  - Gantt of affected flights or processes, if delay is expected (information level 1, 2, 3)
  - Bus departure/arrival times from/to the terminal (information level 0, 1, 2, 3)
  - List of problematic passengers sorted by the EOBT of the connecting flight (information level 0, 1, 2, 3):
    - Name
    - Class
    - If transfer, arrival flight number
    - If transfer, arrival flight AIBT and gate
    - Last-seen info
    - Bag tags of checked-in bags
    - Nature of problem (missing, visa, etc.)
- Start-up tab
  - Stand number (information level 0)
  - E/AOBT, C/ATOT (information level 0)
  - RSAT (information level 0)
  - ASAT (information level 0)
  - Aircraft ready (information level 0)
  - Remove push-back (information level 0)
- De-icing (out of scope for the verification scenario)