



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΠΛΗΡΟΦΟΡΙΑΚΑ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

SmartFalcon: «Ανάπτυξη Αυτόματου Συστήματος
Αξιολόγησης Ιπταμένων σε Εξομοιωτή Πτήσεων
Αεροσκαφών General Dynamics F-16»

SmartFalcon: “Development of an Automated Pilot
Evaluation System for the General Dynamics F-16
Flight Simulator”

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Των

Γεώργιου Σωτηρίου – icsd518024
Δημήτριου Τζάλλα –icsd518025

Επιβλέπων : Ιωάννης Χαραλαμπίδης, Καθηγητής

Μέλη εξεταστικής επιτροπής: Ευριπίδης Λουκής, Καθηγητής
Χρήστος Γκουμόπουλος, Αναπληρωτής Καθηγητής

Σάμος, Ιούνιος 2020

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον διδάσκοντα και επιβλέποντά μας Καθηγητή Ιωάννη Χαραλαμπίδη για την ευκαιρία που μας έδωσε να ασχοληθούμε με το ενδιαφέρον αυτό θέμα και την καθοδήγηση που παρείχε, καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μας εργασίας.

Οφείλουμε να ευχαριστήσουμε όλους τους Ιπταμένους, οι οποίοι έλαβαν μέρος στο ερωτηματολόγιο, το οποίο εκπονήθηκε για τους σκοπούς έρευνας στα πλαίσια αυτής της διπλωματικής εργασίας.

Ευχαριστούμε επίσης, τις οικογένειες μας για την υπομονή τους και την αδιάλειπτη υποστήριξη που μας παρείχαν, σε όλες τις φάσεις της συγγραφής της εργασίας και των σπουδών μας.

© 2020

Με επιφύλαξη παντός δικαιώματος.

ΣΩΤΗΡΙΟΥ ΓΕΩΡΓΙΟΣ

ΤΖΑΛΛΑΣ ΔΗΜΗΤΡΙΟΣ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας περιεχομένων

1	Εισαγωγή	1
1.1	Ζητήματα προσαρμογής των ιπταμένων στους εξομοιωτές πτήσεων	1
1.2	Δομή διπλωματικής εργασίας	3
1.3	Στόχοι	3
2	Απαιτήσεις Εκπαίδευσης ιπτάμενου προσωπικού Πολεμικής Αεροπορίας στον εξομοιωτή πτήσεων.	5
2.1	Εισαγωγή	5
2.2	Πληθυσμός και Δείγμα	5
2.3	Ερωτηματολόγιο Έρευνας	6
2.4	Ανάλυση αποτελεσμάτων	9
2.5	Συμπεράσματα Ερωτηματολογίου	17
2.6	Απαιτήσεις Εκπαίδευσης	18
2.7	Πίνακες Απόδοσης	18
3	Προδιαγραφές και απαιτήσεις Smart Falcon	20
3.1	Εισαγωγή	20
3.2	Προδιαγραφές λογισμικού	20
3.3	Διάγραμμα περιπτώσεων χρήσης του Smart Falcon	37
3.4	Μεθοδολογία ανάπτυξης λογισμικού Smart Falcon	38
4	Δημιουργία προγράμματος αξιολόγησης Smart Falcon	40
4.1	Εισαγωγή	40
4.2	Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν	40
4.3	Javascript	41
4.4	React.js	41
4.5	Smart Falcon User Interface	42
4.6	Smart Falcon Functions	46
4.6.1	handleIfTasPassOrFail()	46
4.6.2	handleIfHeadingPassOrFail()	47
4.6.3	handleIfgForcePassOrFail()	48
5	Δοκιμές λογισμικού Smart Falcon	49
5.1	Εισαγωγή	49
5.2	Βήμα 1ο: Πτήση στον εξομοιωτή με το λογισμικό Falcon BMS 4.32	49

5.3	Βήμα 2ο: Χρήση του λογισμικού Tacview	51
5.4	Βήμα 3ο. Χρήση του λογισμικού Smart Falcon	53
5.5	Αποτελέσματα Χρήσης	53
5.5.1	Περίπτωση Α	53
5.5.2	Περίπτωση Β	54
6	Viper Training Station	56
6.1	Εισαγωγή	56
6.2	Κατηγοριοποίηση flight simulator σύμφωνα με τον ICAO.	56
6.3	Κατηγοριοποίηση του Viper Training Center	58
6.4	Περιγραφή Viper Training Center	58
6.4.1	Cockpit	58
6.4.2	Οθόνι	59
6.4.3	Διασύνδεση	59
6.4.4	Tactical Controller Computer	60
6.4.5	Debriefing Station Computer	60
6.4.6	Flight Simulator Server	60
6.4.7	Πλεονεκτήματα και μειονεκτήματα	61
7	Οικονομική Μελέτη	62
7.1	Εισαγωγή	62
7.2	Κόστος χρήσης υπαρχόντων flight simulator	62
7.3	Κόστος εγκατάστασης Viper Training Center	62
7.4	Κόστος χρήσης Viper Training Center	63
8	Προοπτικές-Συμπεράσματα	65
8.1	Προοπτικές	65
8.2	Συμπεράσματα	65
	Βιβλιογραφία	67

Πίνακας Εικόνων

Εικόνα 1: Πτητική εμπειρία	10
Εικόνα 2: Εκπαιδευτής πτήσεων	10
Εικόνα 3: Χαρακτηρισμός πρότερης εκπαίδευσης στο flight simulator	11
Εικόνα 4: Σημαντικότερος τομέας συνεισφοράς εκπαίδευσης στο flight simulator	11
Εικόνα 5: Άγχος από την ύπαρξη εκπαιδευτή στο flight simulator.....	12
Εικόνα 6: Σημαντικότητα του εκπαιδευτή στο flight simulator	13
Εικόνα 7: Αντικειμενικότητα αξιολόγησης εκπαιδευτών στο flight simulator	13
Εικόνα 8: Δυνατότητα απενημέρωσης από τα υπάρχοντα flight simulator	14
Εικόνα 9: Συχνότητα εκτέλεσης απενημέρωσης πτήσεων στο flight simulator.....	14
Εικόνα 10: Ρεαλιστικότητα των κωδικών επίδοσης.....	15
Εικόνα 11: Αντικατάσταση εκπαιδευτή στο flight simulator.....	15
Εικόνα 12: Αντικειμενικότητα αξιολόγησης από λογισμικό.....	16
Εικόνα 13: Διασύνδεση flight simulator	16
Εικόνα 14: Προτίμηση ανάμεσα σε εκπαιδευτή και λογισμικό	17
Εικόνα 15: Use case diagram Smart Falcon.....	37
Εικόνα 16: Μέθοδος Πρωτοτυποποίησης	39
Εικόνα 17: SmartFalconUserInterface	42
Εικόνα 18: Κώδικας δημιουργίας πεδίου TAS-MAX.....	42
Εικόνα 19: Κώδικας για την δημιουργία του πεδίου Upload	44
Εικόνα 20: Πεδίο Time Start	45
Εικόνα 21: Κώδικας για την δημιουργία πίνακα αποτελεσμάτων.....	46
Εικόνα 22: Κώδικας της συνάρτησης handleIfTasPassOrFail()	47
Εικόνα 23: Κώδικας της συνάρτησης handleIfHeadingPassOrFail().....	47
Εικόνα 24: Κώδικας της συνάρτησης handleIfgForcePassOrFail()	48
Εικόνα 25: Ρεαλιστικότητα Falcon BMS	50
Εικόνα 26: AVTR F-16 (Airborne Video Tape Recorder).....	50
Εικόνα 27: Άνοιγμα αρχείου example.vhs με το Tacview	51
Εικόνα 28: Λειτουργία Telemetry Export στο Tacview.....	52
Εικόνα 29: Ολοκλήρωση διαδικασίας Telemetry Export στο Tacview	52
Εικόνα 30: Step 1 και Step 2 στο Smart Falcon	53
Εικόνα 31: Προς εξέταση παράμετροι (Περίπτωση Α).....	54
Εικόνα 32: Αποτελέσματα αξιολόγησης από το Smart Falcon (Περίπτωση Α).....	54
Εικόνα 33: Προς εξέταση παράμετροι (Περίπτωση Β).....	55
Εικόνα 34: Αποτελέσματα αξιολόγησης από το Smart Falcon (Περίπτωση Β).....	55
Εικόνα 35: Cockpit Viper Training Center	58
Εικόνα 36: Οθόνη Viper Training Center	59

Λίστα Πινάκων

Πίνακας 1: Κώδικες απόδοσης.....	19
Πίνακας 2: Κατηγοριοποίηση flight simulator (ICAO)	57
Πίνακας 3: Εκτιμώμενο κόστος Viper Training Center	63

Ακρωνύμια

ICAO	International Civil Aviation Organization
FSTD	Flight Simulator Training Device
ATC	Air Traffic Control
CSV File	Comma Separated Values File
AVTR	Audio Visual Tape Recorder

Σύνοψη

Οι εξομοιωτές πτήσεων αποτελούν ένα αναπόσπαστο τμήμα της εκπαίδευσης των ιπταμένων. Εξελίσσονται διαρκώς και αντικαθιστούν ολοένα και περισσότερο την πραγματική πτητική εκπαίδευση, μειώνοντας δραματικά το κόστος και τους κινδύνους της. Στο πρώτο τμήμα της παρούσας εργασίας, αναδεικνύονται οι αδυναμίες των υπάρχοντων συστημάτων εξομοίωσης πτήσης, αξιολογώντας τις απαντήσεις 52 ιπταμένων σε δοθέν ερωτηματολόγιο. Στην συνέχεια προδιαγράφεται και εφαρμόζεται το Smart Falcon, ένα λογισμικό αυτόματης αξιολόγησης των ιπτάμενων που συνεργάζεται με open source εξομοιωτές πτήσεων και παρουσιάζονται τα αποτελέσματα από την χρήση του. Τέλος, παρουσιάζεται η οικονομική διάσταση του ζητήματος, μέσω της πρότασης δημιουργίας ενός “off the shelf” συστήματος εξομοίωσης και αναδεικνύονται η δυνατότητες αναβάθμισής τους.

Λέξεις Κλειδιά: Εξομοιωτής πτήσεων, Αεροπορία, Εκπαίδευση Ιπταμένων, Αυτόματο σύστημα.

Abstract

Flight simulators are an integral part of a pilot's training. They are constantly evolving and replacing the real world flight training, thus reducing dramatically its cost and risks. In the first part of this thesis, we are highlighting the weaknesses of the existing flight simulator systems, by analyzing the answers of 52 pilots in a given questionnaire. Next, we specify and develop the Smart Falcon, an automated pilot evaluation software, that is applied on an open source flight simulator and we present the results of its use. Finally, we propose the creation of an “off the self” flight simulator system, present the economic side of its use and highlight its upgrade potential.

Key Words: Flight simulator, Aviation, Pilot training, Automated System

1

Εισαγωγή

1.1 Ζητήματα προσαρμογής των ιπταμένων στους εξομοιωτές πτήσεων

Η εκπαίδευση ενός ιπτάμενου της Πολεμικής Αεροπορίας είναι ένα πολύ απαιτητικό έργο με μεγάλο κόστος τόσο σε πόρους όσο και σε χρόνο. Από την έναρξη της εκπαίδευσης έως και το στάδιο ενός πλήρως ετοιμοπόλεμου αεροπόρου, απαιτείται η συνδυασμένη προσπάθεια πολλών εκπαιδευτών και προσωπικού διαφόρων ειδικοτήτων, ενώ έχει υπολογιστεί ότι το κόστος ανέρχεται σε μερικές δεκάδες εκατομμύρια ευρώ. Διαχρονικά γίνεται προσπάθεια για αύξηση της αποτελεσματικότητας της παρεχόμενης εκπαίδευσης με παράλληλη μείωση του κόστους. Για το λόγο αυτό ένα μεγάλο μέρος της εκπαίδευσης λαμβάνει χώρα στους εξομοιωτές πτήσεων.

Οι εξομοιωτές είναι συσκευές οι οποίες μιμούνται τα συστήματα του αεροσκάφους και τις συνθήκες πτήσης προκειμένου να προσφέρουν το μέγιστο δυνατό ρεαλισμό στην εκπαίδευση ενός ιπτάμενου με πολύ μικρό κόστος ανά ώρα πτήσης. Η εκπαίδευση των αεροπόρων τόσο στην πολιτική όσο και στην Πολεμική Αεροπορία στρέφονται ολοένα και περισσότερο στην χρήση εξομοιωτών πτήσεων, λόγω της τεχνολογικής ανάπτυξης στον τομέα του ρεαλισμού. Ταυτόχρονα, με την αύξηση της εκπαίδευσης σε εξομοιωτές, αυξάνονται και οι ανάγκες σε εκπαιδευτές. Όπως σε κάθε σύστημα, έτσι και στους εξομοιωτές πτήσεων η ύπαρξη του παράγοντα “άνθρωπος” (εκπαιδευτής) παρουσιάζει μειονεκτήματα και πλεονεκτήματα. Το σημαντικότερο από τα πλεονεκτήματα είναι ότι ο εκπαιδευτής μπορεί να μεταδώσει την εμπειρία του στο μαθητή μέσα από επαναλαμβανόμενα σενάρια εκπαίδευσης στον εξομοιωτή. Αντίθετα, λόγω περιορισμών της φύσης του ανθρώπου, ένας εκπαιδευτής είναι πρακτικά αδύνατον να παρακολουθεί ταυτόχρονα την εξέλιξη του πλήθους των μεταβλητών που μεταβάλλονται κατά τη διάρκεια μιας πτήσης όπως το ύψος, η ταχύτητα και η πορεία του

αεροσκάφους. Αποτέλεσμα αυτού είναι ο εκπαιδευτής να μην μπορεί να παρέχει τη μέγιστη δυνατή εκπαίδευση διορθώνοντας όλα τα σφάλματά του εκπαιδευόμενου. Παράλληλα, παράγοντες όπως η ψυχολογία, το προσωπικό στυλ διδασκαλίας και ικανότητα επικοινωνίας του κάθε εκπαιδευτή επηρεάζει με ανόμοιο τρόπο την εκπαίδευση των ιπταμένων. Επιπλέον απαιτούνται πολλές εργατοώρες για την ολοκλήρωση ενός προγράμματος εκπαίδευσης στον εξομοιωτή, με αποτέλεσμα τις περισσότερες φορές να υπάρχει υπερβολικός φόρτος εργασίας στους εκπαιδευτές λόγω των ταυτόχρονων πτητικών υποχρεώσεων τους. Στην Πολεμική Αεροπορία ο φόρτος αυξάνεται ακόμα περισσότερο λόγω των επιχειρησιακών απαιτήσεων.

Τα προβλήματα αυτά είναι ήδη γνωστά από τη δεκαετία του '80 και ήδη από την δεκαετία του '90 έγιναν προσπάθειες για την αφαίρεση του παράγοντα “άνθρωπος” από το σύστημα εξομοιωτής-εκπαιδευόμενος μέσω της εισαγωγής έξυπνων λογισμικών αξιολόγησης και διδασκαλίας των ιπτάμενων στους υπάρχοντες εξομοιωτές πτήσεων. Η έρευνα και ανάπτυξη έγινε κυρίως από εταιρείες που παρήγαγαν συστήματα εξομοίωσης σε συνεργασία με πανεπιστημιακούς φορείς. Ένα τέτοιο παράδειγμα είναι το λογισμικό TacAir-Soar που αναπτύχθηκε στο εργαστήριο WISSARD στην Ναυτική Βάση της Oceana και το Εργαστήριο Ερευνών της USAF στην Αριζόνα. Το λογισμικό αυτό έχει την δυνατότητα να εξομοιώσει την συμπεριφορά των ιπτάμενων κατά την πτήση σε πολεμικές αποστολές και έτσι ο εκπαιδευόμενος εξασκείται σε περιβάλλον με ρεαλιστικούς αντιπάλους (Jones, et al., 1999). Ένα άλλο παρόμοιο σύστημα αναπτύχθηκε τη δεκαετία του '90 από τον στρατό των Ηνωμένων Πολιτειών με σκοπό τη βελτίωση της εκπαίδευσης των εκπαιδευομένων στα ελικόπτερα. Το συγκεκριμένο σύστημα μπορεί να αξιολογεί την απόδοση του εκπαιδευόμενου και να προσαρμόζει ανάλογα το επίπεδο δυσκολίας, παρέχοντας έτσι εξατομικευμένη εκπαίδευση (Remolina, Ramachandran, Fu, Stottler, &Howse, 2004). Αυτές οι προσπάθειες εξέλιξης των συστημάτων εξομοίωσης υποδηλώνουν την αναγκαιότητα για διαρκή μείωση του κόστους εκπαίδευσης με ταυτόχρονη αύξηση της αποτελεσματικότητάς της.

Παρόλο που τα παραπάνω συστήματα εξομοίωσης λύνουν σε μεγάλο βαθμό τα προβλήματα που δημιουργούνται από τον παράγοντα “άνθρωπος”, έχουν συνήθως μεγάλο κόστος, σαφώς όμως μικρότερο από εκείνο μιας πραγματικής πτήσης. Η μεγάλη ανάπτυξη όμως που έχει παρατηρηθεί στον τομέα των simulation games, παρέχει την δυνατότητα χρήσης τους ως μια οικονομική πλατφόρμα εξομοίωσης. Οι εξομοιωτές πτήσεων όπως το Microsoft Flight Simulator, το DCS World και το Falcon BMS προσφέρουν υψηλό ρεαλισμό που σε πολλές περιπτώσεις πλησιάζει αυτόν ενός πραγματικού εξομοιωτή. Το γεγονός αυτό σε συνδυασμό με το χαμηλό κόστος και τον open source χαρακτήρα πολλών gaming εξομοιωτών (Falcon BMS), δίνει τη δυνατότητα για τη δημιουργία ενός εξαιρετικού υποστηρικτικού εργαλείου εκπαίδευσης ιπτάμενων.

Κατανοώντας την αναγκαιότητα αύξησης της αποτελεσματικότητας της εκπαίδευσης των ιπταμένων, στη διπλωματική εργασία αυτή γίνεται προσπάθεια δημιουργίας του Smart Falcon, ενός αυτόματου συστήματος αξιολόγησης ιπτάμενων στον εξομοιωτή πτήσεων. Προσπαθώντας να εξαλείψουμε τα μειονεκτήματα από την ύπαρξη του ανθρώπινου παράγοντα στον κύκλο εκπαίδευσης και να μειώσουμε το κόστος, προσθέτουμε μία web-based εφαρμογή του σε έναν open source εξομοιωτή πτήσεων και παρουσιάζουμε την πρακτική και οικονομική διάσταση του ζητήματος.

1.2 Δομή διπλωματικής εργασίας

Η παρούσα διπλωματική εργασία χωρίζεται σε οκτώ κεφάλαια. Συγκεκριμένα στο δεύτερο κεφάλαιο (Απαιτήσεις Εκπαίδευσης ιπτάμενου προσωπικού Πολεμικής Αεροπορίας στον εξομοιωτή πτήσεων) γίνεται έρευνα με την μορφή ερωτηματολογίου σχετικά την παρούσα εκπαίδευση στον εξομοιωτή πτήσεων και το κατά πόσο οι ιπτάμενοι θεωρούν απαραίτητη την εισαγωγή λογισμικού στην θέση του εκπαιδευτή στον εξομοιωτή πτήσεων.

Στο τρίτο κεφάλαιο (Προδιαγραφές και απαιτήσεις Smart Falcon) ορίζονται οι προδιαγραφές και οι απαιτήσεις του λογισμικού Smart Falcon το οποίο προτείνεται στην παρούσα διπλωματική για την υποβολή, ακόμη και την αντικατάσταση του εκπαιδευτή πτήσεων στον flight simulator.

Στο τέταρτο κεφάλαιο (Δημιουργία προγράμματος αξιολόγησης SmartFalcon) αναλύεται η δημιουργία του λογισμικού Smart Falcon και αναλύονται το user interface καθώς και τα σημαντικότερα κομμάτια του κώδικά του.

Στο πέμπτο κεφάλαιο (Δοκιμές λογισμικού Smart Falcon) περιγράφεται όλη η διαδικασία την οποία θα πρέπει να εκτελέσει ο χρήστης για να χρησιμοποιήσει το λογισμικό και παρουσιάζονται δύο περιπτώσεις χρήσεις. Στην πρώτη ο εκπαιδευόμενος πετυχαίνει τις παραμέτρους που εξετάζει το λογισμικό και στην δεύτερη αποτυγχάνει.

Στο έκτο κεφάλαιο (Viper Training Station) γίνεται κατηγοριοποίηση του Viper Training Center σύμφωνα με τον ICAO καθώς και λεπτομερής περιγραφή του.

Στο έβδομο κεφάλαιο (Οικονομική Μελέτη) αναλύεται το κόστος χρήσης των υπαρχόντων flight simulator και προσδιορίζεται το κόστος εγκατάστασης και χρήσης του Viper Training Center.

Στο όγδοο και τελευταίο κεφάλαιο (Συμπεράσματα-Προοπτικές) επισημαίνονται τα συμπεράσματα από την χρήση του Smart Falcon καθώς και οι δυνατότητες αναβάθμισης και εξέλιξής του με την εισαγωγή νέων τεχνολογιών.

1.3 Στόχοι

Κύριος αντικειμενικός σκοπός της παρούσας εργασίας είναι να γίνει πλήρως κατανοητή η ανάγκη εξέλιξης των υπαρχόντων συστημάτων εξομοίωσης και τα πολλαπλά οφέλη από την

χρήσης τους. Ως επιμέρους αντικειμενικοί σκοποί ορίζονται οι παρακάτω: α) να γίνει αντιληπτό το γεγονός ότι γίνεται συνεχής προσπάθεια μείωσης του κόστους της εκπαίδευσης των ιταμένων μέσω της χρήσης εξομοιωτών πτήσεων, β) και ταυτόχρονη προσπάθεια αύξησης του ρεαλισμού της εκπαίδευσης, γ) να παρουσιαστούν τα πλεονεκτήματα και τα μειονεκτήματα από την ύπαρξη του ανθρώπινου παράγοντα (εκπαιδευτής πτήσεων) στο σύστημα εξομοιωτής-εκπαιδευόμενος, δ) να παρουσιαστούν τα πλεονεκτήματα από την χρήση αυτόματων συστημάτων αξιολόγησης και συγκεκριμένα από το πρόγραμμα Smart Falcon, ε) να αναδειχθούν τα πλεονεκτήματα και τα μειονεκτήματα από την χρήση λογισμικών αυτόματης αξιολόγησης σε συνδυασμό με open source εξομοιωτές πτήσεων και στ) να προβληθούν οι απεριόριστες δυνατότητες εξέλιξης των εν λόγω συστημάτων μέσω της εφαρμογής νέων τεχνολογιών.

2

Απαιτήσεις Εκπαίδευσης

ιπτάμενου προσωπικού

Πολεμικής Αεροπορίας

στον εξομοιωτή πτήσεων.

2.1 Εισαγωγή

Προκειμένου να αναδείξουμε την ανάγκη δημιουργίας ενός έξυπνου λογισμικού όπως το Smart Falcon, χρησιμοποιήσαμε ένα ερωτηματολόγιο απόψεων. Μέσω αυτού, έγινε προσπάθεια προσδιορισμού των απαιτήσεων εκπαίδευσης του ιπτάμενου προσωπικού της Πολεμικής Αεροπορίας στο flight simulator. Στην ενότητα γίνεται η παρουσίαση του ερωτηματολογίου που χρησιμοποιήθηκε για να διαπιστωθεί κατά πόσο το ιπτάμενο προσωπικό είναι δεκτικό στην αντικατάσταση του εκπαιδευτή πτήσεων από ένα λογισμικό, αλλά και το πόσο ρεαλιστικούς θεωρεί τους κώδικες επίδοσης που χρησιμοποιούνται στην παρούσα εκπαίδευσή του. Παρουσιάζονται τα αποτελέσματα του ερωτηματολογίου, καθορίζονται οι απαιτήσεις εκπαίδευσης στο flight simulator και τέλος παρουσιάζονται οι πίνακες αναμενόμενης απόδοσης των εκπαιδευομένων τους οποίους θα πρέπει να χρησιμοποιεί το λογισμικό ώστε να τους αξιολογεί.

2.2 Πληθυσμός και Δείγμα

Εξαιτίας του γεγονότος ότι η ανάπτυξη του λογισμικού ευφυούς πτητικής εκπαίδευσης στο flight simulator αφορά όλους τους ιπτάμενους που χρησιμοποιούν το

flight simulator είτε είναι εκπαιδευτές είτε εκπαιδευόμενοι, θεωρήθηκε κατάλληλη η προσέγγιση του θέματος από τη σκοπιά και των δύο. Για τον ίδιο λόγο το δείγμα δεν περιορίστηκε μόνο στους ιπτάμενους της Πολεμικής Αεροπορίας αλλά απευθύνθηκε σε ίδιο ποσοστό και σε ιπτάμενους εκτός Πολεμικής Αεροπορίας.

2.3 Ερωτηματολόγιο Έρευνας

Στόχοι της έρευνας που πραγματοποιήθηκε με την βοήθεια τους ερωτηματολογίου ήταν να διαπιστωθούν τα παρακάτω:

1. Η εμπειρία του ιπτάμενου
2. Η ιδιότητά του (εκπαιδευτής πτήσεων και flight simulator).
3. Η γνώμη του για την έως τώρα εκπαίδευσή του στο flight simulator.
4. Η γνώμη του για την αντικατάσταση του ρόλου του εκπαιδευτή από λογισμικό
5. Πιθανές προτάσεις του για βελτίωση των υπαρχόντων flight simulator.

Για να ικανοποιηθούν οι παραπάνω στόχοι του ερωτηματολογίου χρησιμοποιήθηκαν οι παρακάτω ερωτήσεις:

1. Πόση είναι η πτητική σας εμπειρία (συνολικές ώρες πτήσης);

- 0-500 (άπειρος)
- 500-1000 (ενδιάμεσης εμπειρίας)
- 1000+ (έμπειρος)

2. Είστε εκπαιδευτής πτήσεων;

- Ναι.
- Όχι.

3. Πώς θα χαρακτηρίζατε την έως τώρα εκπαίδευσή σας στο flight simulator;

- Πολύ κακή.
- Κακή.
- Μέτρια.
- Καλή.

- Άριστη.

4.Που θεωρείται ότι βοηθά περισσότερο η εκπαίδευση στο flight simulator;

- Διαδικασίες ανάγκης.
- Πτήση δι' οργάνων.
- Πτήσεις σε τακτικό περιβάλλον.

5.Σε ποιόν βαθμό σας αγχώνει η ύπαρξη εκπαιδευτή πτήσεων στο flight simulator;

- Καθόλου.
- Σχεδόν καθόλου.
- Μέτρια.
- Πολύ.
- Πάρα πολύ.

6.Πόσο σημαντικός πιστεύετε ότι είναι ο ρόλος του εκπαιδευτή πτήσεων στο flight simulator;

- Καθόλου.
- Σχεδόν καθόλου.
- Μέτρια.
- Πολύ.
- Πάρα πολύ.

7.Πιστεύεται ότι η αξιολόγηση των πτήσεων σας από τους εκπαιδευτές στο flight simulator είναι αντικειμενική;

- Ναι.
- Όχι.

8.Στο flight simulator που χρησιμοποιείτε παρέχεται η δυνατότητα απενημέρωσης της αποστολής που πετάξατε;

- Ναι.
- Όχι.

9.Πόσο συχνά κάνετε απενημέρωση για τις πτήσεις που πετάξατε στο flight simulator;

- Ποτέ.
- Σχεδόν ποτέ.
- Στο 50% των πτήσεων.
- Σχεδόν πάντα.
- Πάντα.

10.Σε ποιόν βαθμό πιστεύετε ότι οι κώδικες επίδοσης που χρησιμοποιούνται για την εκπαίδευσή σας είναι ρεαλιστικοί;

- Καθόλου.
- Σχεδόν καθόλου.
- Μέτρια.
- Πολύ.
- Πάρα πολύ.

11.Πιστεύετε ότι ο εκπαιδευτής πτήσεων θα μπορούσε να αντικατασταθεί από λογισμικό στα πλαίσια της εκπαίδευσης στο flight simulator;

- Ναι.
- Όχι.

12.Πόσο αντικειμενική πιστεύετε ότι θα ήταν η αξιολόγηση σας από ένα λογισμικό στο flight simulator;

- Καθόλου.
- Σχεδόν καθόλου.
- Μέτρια.
- Πολύ.
- Πάρα πολύ.

13.Διαλέξτε ένα από τα παρακάτω:

- Θα ήθελα μόνο άνθρωπο εκπαιδευτή στο flight simulator.
- Θα ήθελα μόνο λογισμικό για εκπαιδευτή στο flight simulator.

- Θα ήθελα και άνθρωπο και λογισμικό ταυτόχρονα στη θέση του εκπαιδευτή στο flight simulator.

14. Σε ποιόν βαθμό πιστεύετε ότι η διασύνδεση πολλών flight simulator θα ενίσχυε την παρεχόμενη εκπαίδευση;

- Καθόλου.
- Σχεδόν καθόλου.
- Μέτρια.
- Πολύ.
- Πάρα πολύ.

15. Τι άλλο πιστεύετε ότι θα μπορούσε να προστεθεί ή να αλλάξει στο υπάρχον flight simulator που χρησιμοποιείται για την εκπαίδευσή σας;

Μέσω της πρώτης ερώτησης του ερωτηματολογίου εκπληρώνουμε τον πρώτο στόχο του ερωτηματολογίου, βλέπουμε δηλαδή την πτητική εμπειρία του δείγματος. Με την δεύτερη ερώτηση εξετάζουμε την ιδιότητά του. Σημειώνεται ότι άπειροι ιπτάμενοι δεν δύνανται να είναι εκπαιδευτές, ενώ όλοι οι έμπειροι είναι εκπαιδευτές. Ο συνδυασμός των παραπάνω θα βοηθήσει στην ομαδοποίηση των επόμενων ερωτήσεων ανά κατηγορία ιπταμένων. Με τις επόμενες οκτώ ερωτήσεις εξετάζουμε την ποιότητα της παρεχόμενης εκπαίδευσης στο flight simulator ικανοποιώντας τον τρίτο στόχο του ερωτηματολογίου, ο οποίος είναι η αξιολόγηση της παρεχόμενης εκπαίδευσης και το κατά πόσο αυτή χρήζει βελτίωσης. Για την ικανοποίηση του τέταρτου σκοπού του ερωτηματολογίου χρησιμοποιήθηκαν τέσσερις ερωτήσεις. Τέλος με την 15η ερώτηση δόθηκε η δυνατότητα για προτάσεις, οι οποίες θα μπορούσαν να βελτιώσουν την εκπαίδευση στο υπάρχον flight simulator.

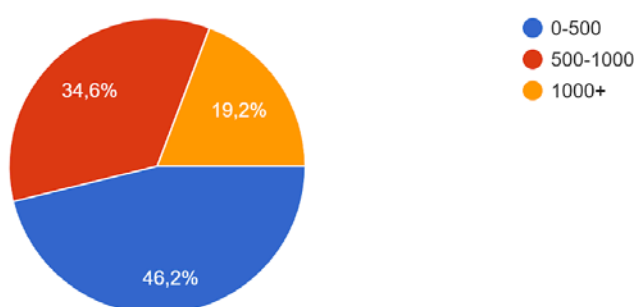
2.4 Ανάλυση αποτελεσμάτων

Το ερωτηματολόγιο συμπληρώθηκε από 52 ιπτάμενους της Πολεμικής αλλά και της Πολιτικής Αεροπορίας χρησιμοποιώντας την εφαρμογή Google Forms και την αποστολή

του μέσω email. Τα αποτελέσματα από την ανάλυση των απαντήσεων παρουσιάζονται στη συνέχεια.

Η πρώτη ερώτηση αφορά την πτητική εμπειρία των ιπτάμενων που απάντησαν στο ερωτηματολόγιο. Από το σύνολο των 52 ερωτηθέντων το 46,2% είναι άπειροι ιπτάμενοι με κάτω από 500 ώρες πτήσης, το 34,6% είναι ιπτάμενοι ενδιάμεσης εμπειρίας και το 19,2% είναι έμπειροι ιπτάμενοι με 1000+ ώρες πτήσης (εικόνα 1). Τα αποτελέσματα αυτά θα τα χρησιμοποιήσουμε στη συνέχεια για να δούμε πώς απάντησε η κάθε μία κατηγορία από αυτές στις ερωτήσεις που ακολούθησαν.

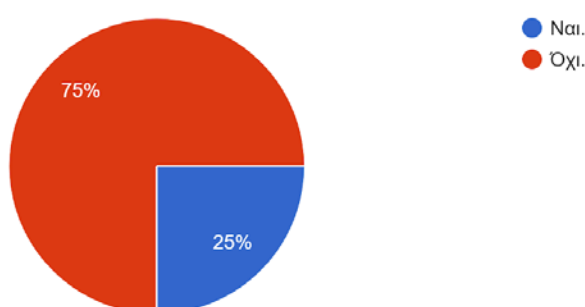
Πόση είναι η πτητική σας εμπειρία (συνολικές ώρες πτήσης);
52 απαντήσεις



Εικόνα 1: Πτητική εμπειρία

Η δεύτερη ερώτηση μας βοηθά να ταξινομήσουμε τους ιπταμένους σε εκπαιδευτές πτήσεων και εκπαιδευόμενους. Αξίζει να αναφέρουμε σε αυτό το σημείο ότι η ιδιότητα του εκπαιδευτή δεν σημαίνει αυτόματα ότι διακόπτεται η εκπαίδευσή του στο flight simulator καθώς η εκπαίδευση ενός ιπτάμενου είναι διαρκής συνεχώς αξιολογούμενη. Από το σύνολο των ερωτηθέντων το 25% είναι εκπαιδευτές ενώ το 75% είναι εκπαιδευόμενοι (Εικόνα 2).

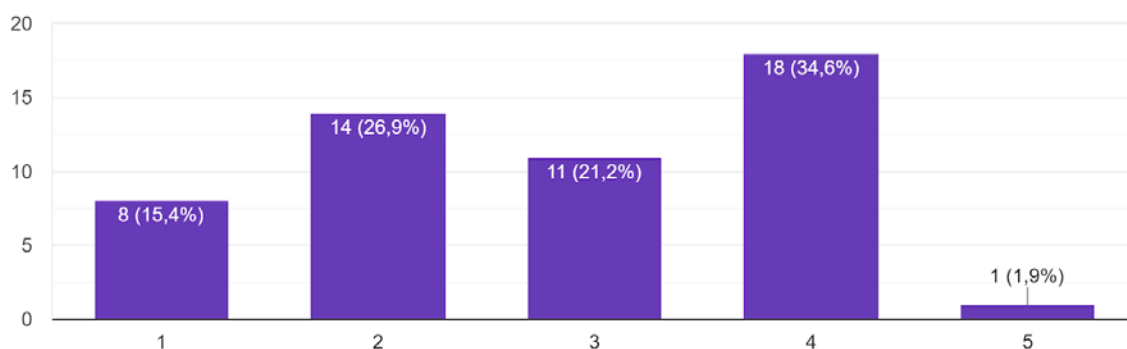
Είστε εκπαιδευτής πτήσεων;
52 απαντήσεις



Εικόνα 2: Εκπαιδευτής πτήσεων

Στην τρίτη ερώτηση, στην οποία ζητήθηκε από τους ιπτάμενους να αξιολογήσουν ποσοτικά το επίπεδο της έως τώρα παρεχόμενης εκπαίδευσης στο flight simulator από πολύ κακή έως άριστη οι απαντήσεις κυμαίνονται σε όλο το εύρος σχεδόν ισόποσα με εξαίρεση την άριστη εκπαίδευση όπου υπήρξε μόνο μία απάντηση (Εικόνα 3). Γίνεται εύκολα αντιληπτό ότι το σύνολο των ιπτάμενων θεωρεί ότι υπάρχει περιθώριο για βελτιώσεις στην παρεχόμενη εκπαίδευση.

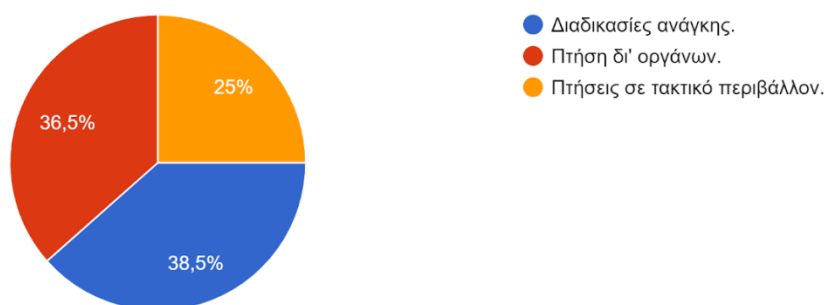
Πώς θα χαρακτηρίζατε την έως τώρα εκπαίδευσή σας στο flight simulator;
52 απαντήσεις



Εικόνα 3: Χαρακτηρισμός πρότερης εκπαίδευσης στο flight simulator

Εξετάζοντας την τέταρτη ερώτηση που διερευνά σε ποιον τομέα βοηθάει περισσότερο η εκπαίδευση στο flight simulator, οι απαντήσεις ήταν σχεδόν ισόποσες και στις τρεις πιθανές απαντήσεις (Εικόνα 4). Από αυτό συμπεραίνουμε ότι το flight simulator είναι σημαντικό τόσο στην εκπαίδευση αντιμετώπισης διαδικασιών ανάγκης, πτήσης δι' οργάνων και πτήσης σε τακτικό περιβάλλον.

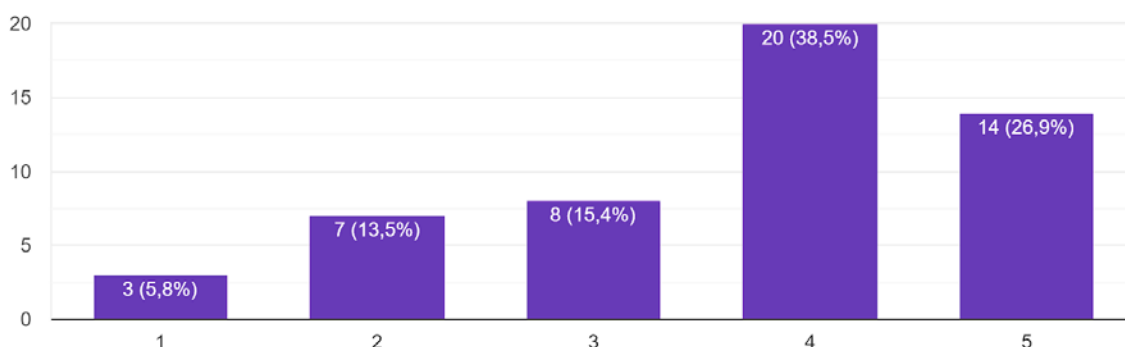
Που θεωρείτε ότι βοηθά περισσότερο η εκπαίδευση στο flight simulator;
52 απαντήσεις



Εικόνα 4: Σημαντικότερος τομέας συνεισφοράς εκπαίδευσης στο flight simulator

Από τις απαντήσεις στην πέμπτη ερώτηση (Εικόνα 5) φαίνεται ότι το 68,4% αγχώνεται πολύ έως πάρα πολύ από την ύπαρξη εκπαιδευτή στην κονσόλα χειρισμού του εξομοιωτή. Είναι εύκολα κατανοητό ότι το άγχος επηρεάζει σε μικρό ή μεγάλο βαθμό την απόδοση των εκπαιδευόμενων ακόμα και σε ένα πλήρως ελεγχόμενο περιβάλλον όπως το flight simulator.

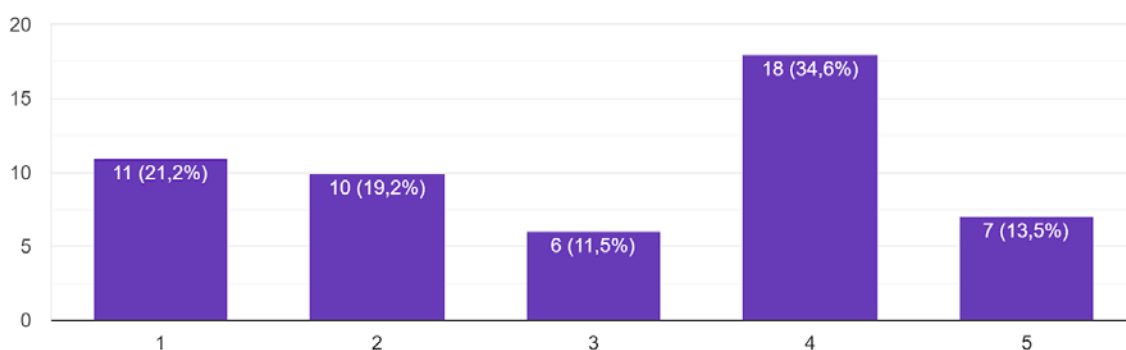
Σας αγχώνει η ύπαρξη εκπαιδευτή πτήσεων στο flight simulator;
52 απαντήσεις



Εικόνα 5: Άγχος από την ύπαρξη εκπαιδευτή στο flight simulator

Στις απαντήσεις της έκτης ερώτησης (Εικόνα 6) βλέπουμε ότι παρόλο που η ύπαρξη εκπαιδευτή αγχώνει τους περισσότερους από τους εκπαιδευόμενους, το 48,1% αναγνωρίζει ότι ο ρόλος του εκπαιδευτή είναι πολύ έως πάρα πολύ σημαντικός για την εκπαίδευσή του. Μια πιθανή εξήγηση είναι ότι παρόλο που υπάρχει η δυνατότητα για εκπαίδευση χωρίς την παρουσία κάποιου εκπαιδευτή, οι ιπτάμενοι θεωρούν πολύ σημαντική την διαρκή αξιολόγησή και την μετάδοση εμπειριών από τους εκπαιδευτές. Άλλωστε η δημιουργία σεναρίων εκπαίδευσης στον εξομοιωτή απαιτεί εμπειρία τόσο πτητική όσο και εκπαιδευτική.

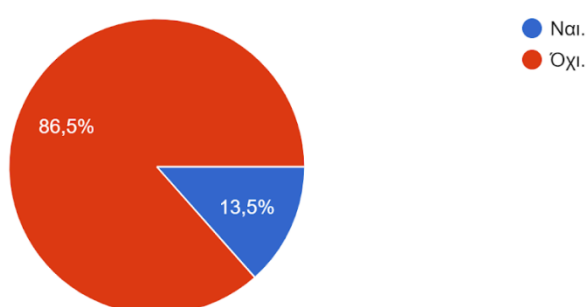
Πόσο σημαντικός πιστεύεται ότι είναι ο ρόλος του εκπαιδευτή πτήσεων στο flight simulator;
52 απαντήσεις



Εικόνα 6: Σημαντικότητα του εκπαιδευτή στο flight simulator

Στην έβδομη ερώτηση (Εικόνα 7) βλέπουμε την συντριπτική πλειοψηφία των ερωτηθέντων (86,5%) να θεωρεί ότι η αξιολόγηση των πτήσεων του στο flight simulator από τους εκπαιδευτές δεν είναι αντικειμενική. Συγκρίνοντας το ποσοστό αυτό με το ποσοστό των ερωτηθέντων που είναι εκπαιδευτές πτήσεων (25%), εύκολα καταλαβαίνουμε ότι και η συντριπτική πλειοψηφία των ίδιων των εκπαιδευτών δεν θεωρεί την αξιολόγηση αντικειμενική.

Πιστεύετε ότι η αξιολόγηση των πτήσεων σας από τους εκπαιδευτές στο flight simulator είναι αντικειμενική;
52 απαντήσεις

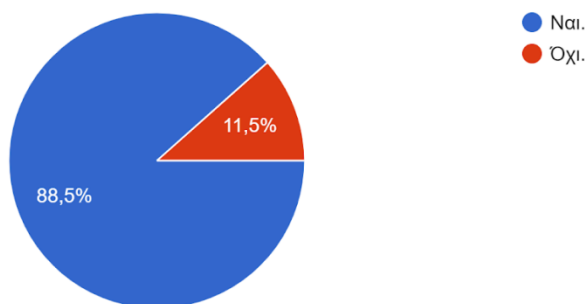


Εικόνα 7: Αντικειμενικότητα αξιολόγησης εκπαιδευτών στο flight simulator

Από τις απαντήσεις στην όγδοη ερώτηση (Εικόνα 8) καταλαβαίνουμε την αναγκαιότητα εκσυγχρονισμού των flight simulator που χρησιμοποιούνται έως σήμερα, καθώς υπάρχουν ακόμη συστήματα (11,5%) τα οποία δεν παρέχουν την δυνατότητα απενημέρωσης των εκπαιδευτικών πτήσεων. Επιπρόσθετα, αξίζει να σημειωθεί ότι η

δυνατότητα απενημέρωσης περιορίζεται στην απλή αναπαραγωγή της πτήσης μετά το πέρας της εκπαίδευσης.

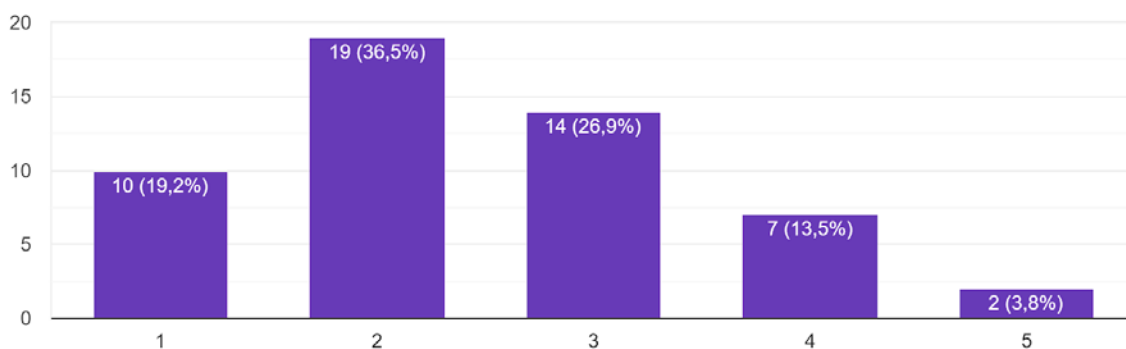
Στο flight simulator που εκπαιδεύεστε παρέχεται η δυνατότητα απενημέρωσης της αποστολής που πετάξατε;
52 απαντήσεις



Εικόνα 8: Δυνατότητα απενημέρωσης από τα υπάρχοντα flight simulator

Στην ένατη ερώτηση (Εικόνα 9) βλέπουμε ότι ακόμη και σε flight simulator που δίνουν τη δυνατότητα απενημέρωσης, αυτή δεν γίνεται στις περισσότερες περιπτώσεις. Ένας από τους κυριότερους λόγους για τον οποίο συμβαίνει αυτό είναι γιατί τα συστήματα απενημέρωσης είναι δύσχρηστα και χρονοβόρα.

Πόσο συχνά κάνετε απενημέρωση για τις πτήσεις που πετάξατε στο flight simulator;
52 απαντήσεις

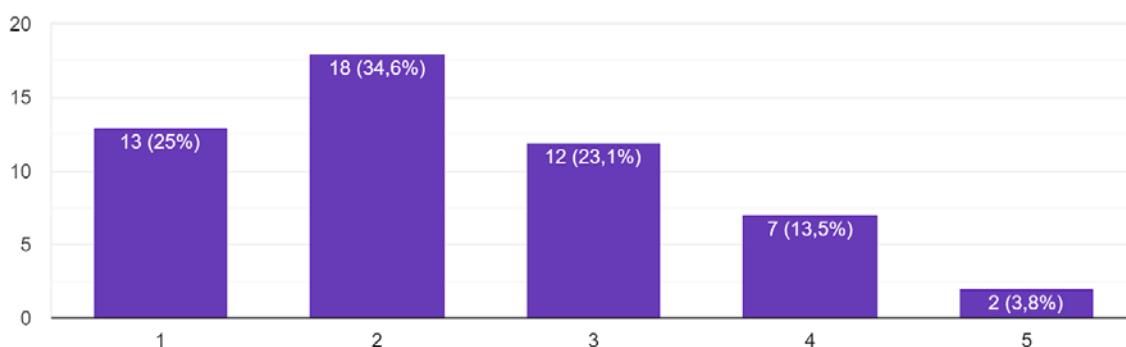


Εικόνα 9: Συχνότητα εκτέλεσης απενημέρωσης πτήσεων στο flight simulator

Στην δέκατη ερώτηση (Εικόνα 10) εξετάζεται το κατά πόσο οι κώδικες επίδοσης που χρησιμοποιούνται για την αξιολόγηση των εκπαιδευομένων στο flight simulator αξιολογούνται ως ρεαλιστικοί. Το 59,6% τους χαρακτηρίζει ως καθόλου ή σχεδόν καθόλου αξιόπιστους. Αυτό μας δείχνει ότι η συντριπτική πλειοψηφία των

εκπαιδευομένων αλλά και των εκπαιδευτών πιστεύει ότι η αξιολόγηση των πτήσεων γίνεται με βάση μη ρεαλιστικούς στόχους στο flight simulator και ότι υπάρχει η ανάγκη για συνεχή αναπροσαρμογή των κωδικών επίδοσης.

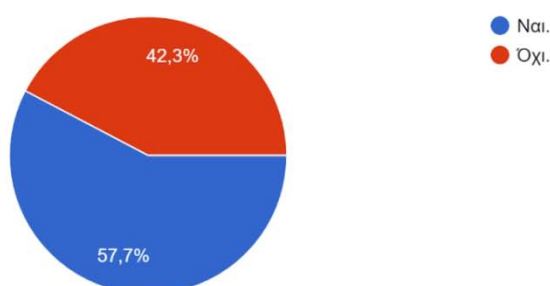
Σε ποιο βαθμό πιστεύετε ότι οι κώδικες επίδοσης που χρησιμοποιούνται για την εκπαίδευσή σας είναι ρεαλιστικοί;
52 απαντήσεις



Εικόνα 10: Ρεαλισμός των κωδικών επίδοσης

Στην ενδέκατη ερώτηση (Εικόνα 11) και στο κατά πόσο θα μπορούσε να αντικατασταθεί ο εκπαιδευτής πτήσεων από λογισμικό στην εκπαίδευση του flight simulator, παρατηρείται τάση υπέρ της χρήσης software στην εκπαίδευση.

Πιστεύετε ότι ο εκπαιδευτής πτήσεων θα μπορούσε να αντικατασταθεί από λογισμικό στα πλαίσια της εκπαίδευσης στο flight simulator;
52 απαντήσεις

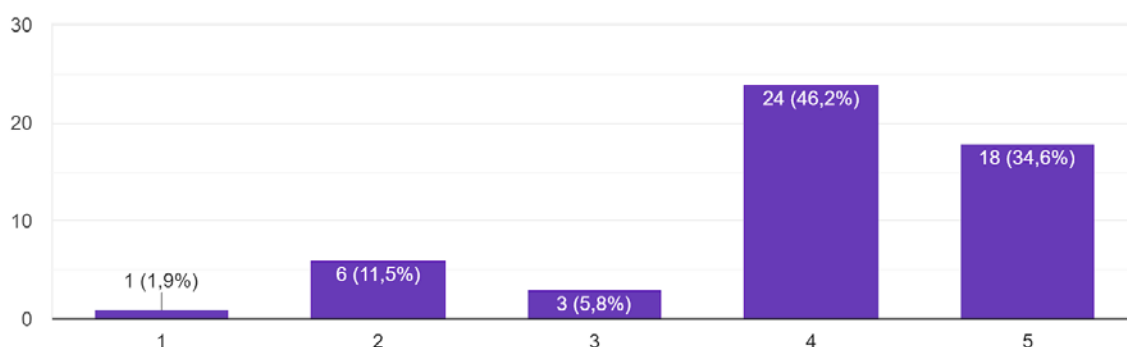


Εικόνα 11: Αντικατάσταση εκπαιδευτή στο flight simulator

Όσον αφορά στο πόσο αντικειμενική θα θεωρούνταν η αξιολόγηση από λογισμικό στα πλαίσια των πτήσεων του flight simulator τα πράγματα είναι πιο ξεκάθαρα (Εικόνα 12). Το 80,8% των ερωτηθέντων στην ερώτηση 12 απάντησε ότι η αξιολόγηση από λογισμικό θα ήταν πολύ έως πάρα πολύ αντικειμενική.

Πόσο αντικειμενική πιστεύετε ότι θα ήταν η αξιολόγηση σας από ένα λογισμικό στο flight simulator;

52 απαντήσεις

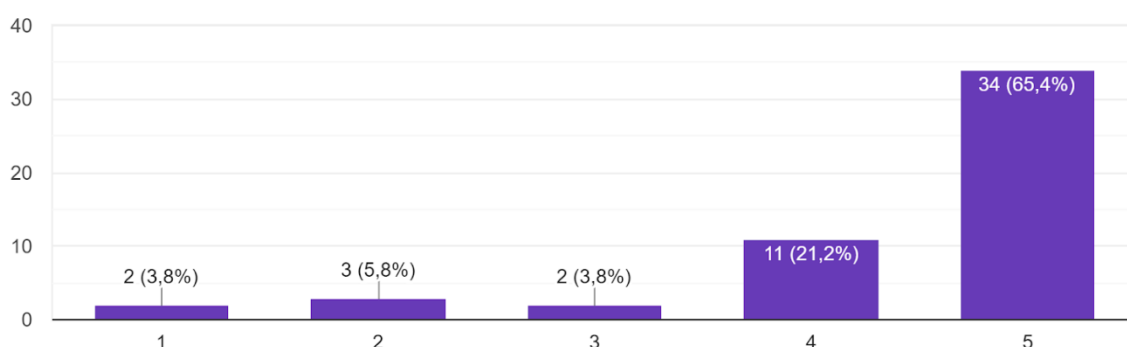


Εικόνα 12: Αντικειμενικότητα αξιολόγησης από λογισμικό

Στην δέκατη τρίτη ερώτηση (Εικόνα 13) εξετάζουμε το κατά πόσο οι ιπτάμενοι θεωρούν σημαντική την διασύνδεση των flight Simulator που χρησιμοποιούν πράγμα το οποίο τους δίνει πολλαπλάσιες δυνατότητες αφού το κάθε simulator μπορεί να αλληλεπιδρά με το άλλο και έτσι να αυξάνεται ο βαθμός ρεαλισμού. Το 86,6% απάντησε ότι θεωρεί πως η διασύνδεση πολλών flight simulator θα ενίσχυε πολύ έως πάρα πολύ την παρεχόμενη εκπαίδευση.

Σε ποιόν βαθμό πιστεύετε ότι η διασύνδεση πολλών flight simulator θα ενίσχυε την παρεχόμενη εκπαίδευση;

52 απαντήσεις



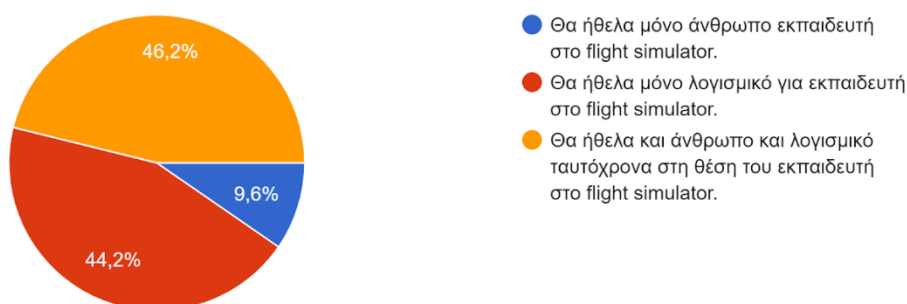
Εικόνα 13: Διασύνδεση flight simulator

Στην προτελευταία ερώτηση (Εικόνα 14) οι ιπτάμενοι κλήθηκαν να απαντήσουν στο εάν στον ρόλο του εκπαιδευτή επιθυμούν μόνο άνθρωπο, μόνο λογισμικό ή την

συνύπαρξη και των δύο. Μόνο το 9,6% απάντησε ότι θα ήθελε να συνεχιστεί η παρούσα κατάσταση με άνθρωπο στον ρόλο του εκπαιδευτή ενώ το 90,4% δήλωσε ότι είναι πρόθυμοι να εκπαιδεύονται από κάποιου είδους λογισμικό ή από συνδυασμό εκπαιδευτή - λογισμικού.

Διαλέξτε ένα από τα παρακάτω:

52 απαντήσεις



Εικόνα 14: Προτίμηση ανάμεσα σε εκπαιδευτή και λογισμικό

Στην τελευταία ερώτηση, η οποία έχει να κάνει με προτάσεις των ιπτάμενων σχετικά με το τι θα μπορούσε να αλλάξει ή να προστεθεί στο υπάρχον flight simulator που χρησιμοποιούν πήραμε τρεις απαντήσεις:

- Καλύτερο γραφικό περιβάλλον.
- Εισαγωγή συστήματος Virtual Reality.
- Περισσότερος ρεαλισμός όπως η εισαγωγή κίνησης στο cockpit.

2.5 Συμπεράσματα Ερωτηματολογίου

Από τις απαντήσεις των ιπτάμενων στο ερωτηματολόγιο διαφαίνεται η ανάγκη για βελτίωση των υπάρχοντων συστημάτων flight simulator. Η βελτίωση αυτή πρέπει να γίνει τόσο στο κομμάτι των εξομοιωτών ως συστήματα, όσο και στον τομέα του τρόπου εκπαίδευσης και αξιολόγησης των ιπτάμενων. Η εξέλιξη των εξομοιωτών στον τομέα της απεικόνισης είναι ένα ζήτημα που απασχολεί τους ιπτάμενους, ενώ ταυτόχρονα η εγκατάσταση νέων τεχνολογιών στους εξομοιωτές (Virtual Reality, Enhanced Graphics) θεωρείται ότι θα λειτουργήσει επιβοηθητικά. Γίνεται κατανοητό ότι ο ανθρώπινος παράγοντας στο ρόλο του εκπαιδευτή διαδραματίζει σημαντικό ρόλο, καθιστώντας μη αντικειμενική την αξιολόγηση των εκπαιδευομένων, ενώ ταυτόχρονα προκαλεί άγχος στους εκπαιδευόμενους. Οι ιπτάμενοι κρίνουν αναγκαία την ύπαρξη κάποιου συστήματος εκπαίδευσης είτε αυτό είναι λογισμικό είτε συνδυασμός ανθρώπου -

λογισμικού. Τέλος κρίνεται αναγκαία η συνεχής βελτίωση στους κώδικες επίδοσης που θα πρέπει να επιτυγχάνονται από τους εκπαιδευόμενους μιας και αυτοί δεν κρίνονται ως ρεαλιστικοί. Οι κώδικες επίδοσης ή αλλιώς πίνακες απόδοσης προκύπτουν από τις απαιτήσεις εκπαίδευσης και με αυτό ακριβώς το κομμάτι θα ασχοληθούμε στη συνέχεια.

2.6 Απαιτήσεις Εκπαίδευσης

Το σύνολο των απαιτήσεων εκπαίδευσης που θα πρέπει να καλυφθούν στην εκπαίδευση ενός ιπτάμενου είναι πολύ μεγάλο και έχει να κάνουν κυρίως με την αποστολή την οποία θα κληθεί να εκτελέσει ο κάθε ιπτάμενος αλλά και το πτητικό μέσο με το οποίο πετάει (μαχητικό, επιβατικό, ελικόπτερο κλπ.). Για παράδειγμα οι απαιτήσεις εκπαίδευσης ενός ιπτάμενου που θα κληθεί να εκτελέσει αποστολές αναχαίτισης με ένα μαχητικό αεροσκάφος είναι τελείως διαφορετικές από τις απαιτήσεις εκπαίδευσης ενός ιπτάμενου ο οποίος θα κληθεί να εκτελέσει μια πτήση μεταφοράς ενός φορτίου. Η παραμετροποίησή τους είναι ένα χρονοβόρο έργο και ξεφεύγει από τα πλαίσια της παρούσας διπλωματικής. Επιπρόσθετα, στοιχεία που αφορούν στην εκπαίδευση των ιπταμένων ειδικά της Πολεμικής Αεροπορίας χαρακτηρίζονται ως απόρρητα και δεν είναι δυνατόν να παρουσιαστούν ως δεδομένα για την υλοποίηση του Smart Falcon. Για αυτό τον λόγο θεωρούμε μια απλοποιημένη εκδοχή της εκπαίδευσης ενός ιπτάμενου. Θα εξετάσουμε μόνο την εκπαίδευση ενός ιπταμένου, ο οποίος, μετά το πέρας της βασικής εκπαίδευσής του, θα κληθεί να εκτελέσει μια πτήση δι' οργάνων με ένα μαχητικό αεροσκάφος.

Η πτήση δι' οργάνων είναι ένα είδος πτήσης που εκτελείται με την χρήση των συστημάτων ναυτιλίας του αεροσκάφους, χωρίς ο ιπτάμενος να βασίζεται σε εξωτερικά σημεία αναφοράς για την εκτέλεση του δρομολογίου του. Συνηθέστερα χρησιμοποιείται όταν οι μετεωρολογικές συνθήκες είναι δυσμενείς (πτήση εντός νεφών, ομίχλη, βροχή κ.α.) και κατά την διάρκεια της νύχτας. Από το πλήθος των παραμέτρων μια τέτοιας πτήσης θα ασχοληθούμε ενδεικτικά με την ικανότητα του ιπτάμενου να διατηρεί σταθερή μια ορισμένη από τους πίνακες απόδοσης ταχύτητα, μια σταθερή πορεία και να μην καταπονεί το αεροσκάφος του με μεγάλες δυνάμεις G , αφού κάτι τέτοιο θα μπορούσε να τον οδηγήσει σε επικίνδυνη κατάσταση.

2.7 Πίνακες Απόδοσης

Με σκοπό να γίνει αντικειμενική η αξιολόγηση των ιπταμένων από τους εκπαιδευτές τους έγινε η εισαγωγή στην αξιολόγηση των ιπταμένων των πινάκων απόδοσης ή αλλιώς κωδικών επίδοσης. Οι πίνακες απόδοσης έκαναν μετρήσιμα τα

βασικά χαρακτηριστικά τα οποία θα πρέπει να ικανοποιεί ο κάθε ιπτάμενος για κριθεί αν είναι ικανός να προχωρήσει στο επόμενο βήμα της εκπαίδευσης ή να επαναλάβει την εκπαίδευσή. Στο παράδειγμα εκπαίδευσης το οποίο θα εξετάσουμε σε αυτή την διπλωματική, ο κάθε εκπαιδευόμενος θα πρέπει να επιτύχει τους κώδικες απόδοσης που φαίνονται στον παρακάτω πίνακα (Πίνακας 1) με τον κώδικα A3 να αποτελεί την χειρότερη απόδοση, η οποία δεν είναι αποδεκτή. Σε αυτήν την περίπτωση ο ιπτάμενος θα πρέπει να επαναλάβει την εκπαίδευσή του. Αντίθετα, ο κώδικας A2 είναι αυτός που θα κρίνει τον εκπαιδευόμενο ως επιτυχών και ο κώδικας A1 είναι αυτός που θα κρίνει το εκπαιδευόμενο ως Άριστο και θα μας δώσει μια ένδειξη ότι μπορούμε να επιταχύνουμε την εκπαίδευσή του. Σε αυτό το σημείο πρέπει να πούμε ότι σε κάθε πτήση ο εκπαιδευόμενος κρίνεται σε όλα τα αντικείμενα εκπαίδευσης και η συνολική βαθμολογία του είναι η ελάχιστη βαθμολογία που έχει επιτύχει. Για παράδειγμα αν ο εκπαιδευόμενος έχει επιτύχει σε όλα τα αντικείμενα εκπαίδευσης κώδικα A1 και σε ένα μόνο αντικείμενο βαθμολογία A2 η συνολική βαθμολογία του είναι A2 και όχι A1 όπως θα μπορούσαμε εύκολα να υποθέσουμε αν υπολογίζαμε τον μέσο όρο της βαθμολογίας του. Ο λόγος που υπάρχει αυτή η αυστηρότητα στην εκπαίδευση είναι ότι, στις πραγματικές πτήσεις τα περιθώρια λάθους είναι πολύ στενά και η πιθανότητα ατυχήματος αυξημένη.

	<u>Κώδικες απόδοσης</u>		
<u>Αντικείμενο Εκπαίδευσης</u>	A3	A2	A1
<u>Διατήρηση Ταχύτητας</u>	±50 Knots	±20 Knots	±10 Knots
<u>Διατήρηση Πορείας Πτήσης</u>	±20°	±10°	±5°
<u>Όρια αεροσκάφους</u>	+2 G's	+1 G's	0 G's

Πίνακας 1: Κώδικες απόδοσης

3

Προδιαγραφές και απαιτήσεις Smart Falcon

3.1 Εισαγωγή

Κάθε λογισμικό που προορίζεται για στρατιωτική χρήση είναι απαραίτητο να έχει συγκεκριμένες προδιαγραφές οι οποίες καθορίζονται από εθνικά και Νατοϊκά πρότυπα. Η απουσία προτύπων στην Πολεμική Αεροπορία έχει ως αποτέλεσμα οι προδιαγραφές αυτές συνήθως πρέπει να εναρμονίζονται με το πρότυπο MIL-STD-498, το οποίο και θα χρησιμοποιηθεί ως βάση για την περιγραφή των απαιτήσεων του λογισμικού μας. Στο συγκεκριμένο κεφάλαιο θα περιγράψουν επίσης οι μη λειτουργικές απαιτήσεις που πρέπει να πληροί το λογισμικό καθώς και οι λειτουργικές απαιτήσεις λαμβάνοντας υπόψη και τα αποτελέσματα του ερωτηματολογίου.

3.2 Προδιαγραφές λογισμικού

Στις παρακάτω προδιαγραφές χρησιμοποιείται ο όρος προγραμματιστής και αναφέρεται στο άτομο ή σύνολο ατόμων, στους οποίους έχει ανατεθεί η δημιουργία του λογισμικού.

Οι γενικές κατευθύνσεις τις οποίες πρέπει να ακολουθήσει κάποιος κατά την ανάπτυξη ενός λογισμικού, συνοψίζονται στα εξής:

- 1) Ο προγραμματιστής θα πρέπει να δημιουργήσει ένα πλάνο ανάπτυξης λογισμικού, το οποίο θα είναι σύμφωνο με τις απαιτήσεις του συμβολαίου. (Department Of Defense, 1994)
- 2) Οι μέθοδοι που θα χρησιμοποιήσει πρέπει να είναι συστηματικές, καταγεγραμμένες και να ακολουθούν το πλάνο που έχει συμφωνηθεί. (Department Of Defense, 1994)
- 3) Κατά τη διάρκεια της ανάπτυξης του λογισμικού, ο προγραμματιστής θα πρέπει να αναγνωρίσει και να αξιολογήσει τμήματα του λογισμικού που μπορούν να

επαναχρησιμοποιηθούν. Τα τμήματα αυτά θα πρέπει να αξιολογηθούν ως προς το κόστος και να επαναχρησιμοποιηθούν όπου κριθεί απαραίτητο.(Department Of Defense, 1994)

- 4) Ο προγραμματιστής πρέπει να αναγνωρίσει τυχόν τμήματα του λογισμικού, των οποίων η βλάβη μπορεί να οδηγήσει σε θάνατο, τραυματισμό, καταστροφή περιουσίας ή περιβαλλοντική καταστροφή. Στην περίπτωση που υπάρχουν τέτοια τμήματα, ο προγραμματιστής θα πρέπει να αναπτύξει μία στρατηγική ασφαλείας έτσι ώστε να μειώσει ή να εξαλείψει την πιθανότητα επικίνδυνων καταστάσεων. Παράλληλα, ο προγραμματιστής θα πρέπει να αναγνωρίσει αν υπάρχουν τμήματα του λογισμικού, των οποίων η βλάβη θα μπορεί να οδηγήσει σε παραβίαση της ασφάλειας του συστήματος. Στην περίπτωση που υπάρχουν τέτοια τμήματα, ο προγραμματιστής θα πρέπει να αναπτύξει μία στρατηγική ασφαλείας με σκοπό να διαβεβαιώσει ότι οι απαιτήσεις, το design, η εφαρμογή και οι διαδικασίες λειτουργίας του συγκεκριμένου τμήματος λογισμικού έχουν ελάχιστες ή μηδαμινές πιθανότητες παραβίασης της ασφάλειας του συστήματος. Τέλος, ο προγραμματιστής θα πρέπει να αναγνωρίσει τυχόν τμήματα του λογισμικού, των οποίων η βλάβη μπορεί να οδηγήσει σε παραβίαση της ιδιωτικότητας του συστήματος. Στην περίπτωση που υπάρχουν τέτοια τμήματα λογισμικού, ο προγραμματιστής θα πρέπει να αναπτύξει μία στρατηγική ασφαλείας της ιδιωτικότητας με σκοπό να διαβεβαιώσει ότι θα εξαλειφθεί ή θα ελαχιστοποιηθεί η πιθανότητα παραβίασης της ιδιωτικότητας του συστήματος.(Department Of Defense, 1994)
- 5) Ο προγραμματιστής θα πρέπει να αναλύσει τις απαιτήσεις σε hardware με σκοπό την εύρυθμη λειτουργία του λογισμικού και να εκτελέσει δοκιμές έτσι ώστε να ικανοποιηθούν όλες οι απαιτήσεις του συμβολαίου.(Department Of Defense, 1994)
- 6) Ο προγραμματιστής θα πρέπει να καταγράψει τη λογική με την οποία πήρε αποφάσεις σε σχέση με τη σχεδίαση, εφαρμογή και δοκιμή του λογισμικού. (Department Of Defense, 1994)
- 7) Ο προγραμματιστής θα πρέπει να παρέχει στον ιδιοκτήτη του λογισμικού ή στον νόμιμο αντικαταστάτη του ελεύθερη πρόσβαση στις εγκαταστάσεις του κατά την διάρκεια ανάπτυξης του λογισμικού. (Department Of Defense, 1994)

Πιο ειδικά το λογισμικό μας θα πρέπει να αναπτυχθεί με βάση τις παρακάτω προδιαγραφές:

- 1) Σχεδιασμός και επίβλεψη του έργου. Ο προγραμματιστής θα πρέπει να εφαρμόσει σχεδιασμό και επίβλεψη του έργου με βάση τις ακόλουθες απαιτήσεις:

- a) Σχεδιασμός ανάπτυξης λογισμικού. Ο προγραμματιστής πρέπει να αναπτύξει και να καταγράψει σχέδια, μέσω των οποίων θα διεξάγει όλες τις δραστηριότητες οι οποίες απορρέουν από το συμβόλαιο. Τα σχέδια αυτά πρέπει να είναι σύμφωνα με τον σχεδιασμό σε επίπεδο συστήματος και πρέπει να περιλαμβάνουν όλα τα πεδία που περιγράφονται στο πλάνο ανάπτυξης λογισμικού.
- b) Σχεδιασμός δοκιμής λογισμικού. Ο προγραμματιστής πρέπει να αναπτύξει και να καταγράψει σχέδια με σκοπό τη διεξαγωγή δοκιμών πιστοποίησης του λογισμικού. Τα σχέδια αυτά πρέπει να περιλαμβάνουν όλα τα πεδία που περιγράφονται στο “Πλάνο δοκιμής του λογισμικού”.
- c) Σχεδιασμός δοκιμής συστήματος. Ο δοκιμαστής πρέπει να συμμετέχει στην ανάπτυξη και καταγραφή σχεδίων, μέσω των οποίων θα διεξαχθεί η δοκιμή και η πιστοποίηση του συστήματος. Τα σχέδια αυτά πρέπει να περιλαμβάνουν όλα τα πεδία που περιγράφονται στο “Πλάνο δοκιμής του λογισμικού”.
- d) Σχεδιασμός εγκατάστασης λογισμικού. Ο προγραμματιστής θα πρέπει να αναπτύξει και να καταγράψει σχέδια, βάσει των οποίων θα εκτελεστεί η εγκατάσταση του λογισμικού και η εκπαίδευση των χρηστών. Τα σχέδια αυτά θα πρέπει να περιλαμβάνουν όλα τα πεδία που περιγράφονται στο “Σχέδιο εγκατάστασης λογισμικού”.
- e) Σχεδιασμός μετάβασης στο νέο λογισμικό. Ο προγραμματιστής θα πρέπει να αναγνωρίσει όλους τους πόρους ανάπτυξης του λογισμικού, οι οποίοι θα χρειαστούν για την υποστήριξη και τη μετάβαση στο νέο λογισμικό. Ο προγραμματιστής θα πρέπει να αναπτύξει και να καταγράψει σχέδια αναγνωρίζοντας αυτούς τους πόρους και περιγράφοντας την προσέγγιση, την οποία πρέπει να ακολουθήσει ο ιδιοκτήτης του λογισμικού για τη μετάβαση στο νέο λογισμικό. Τα σχέδια αυτά πρέπει να περιλαμβάνουν όλα τα πεδία που περιγράφονται στο “Σχέδιο μετάβασης λογισμικού”.
- f) Επακόλουθα σχέδια και σχέδια αναβάθμισης. Μετά την έγκριση όλων των ανωτέρω σχεδίων από τον ιδιοκτήτη του λογισμικού, ο προγραμματιστής θα πρέπει να εκτελέσει όλες οι σχετικές δραστηριότητες σύμφωνα με το πλάνο αυτό. Η διοίκηση της εταιρείας ανάπτυξης λογισμικού πρέπει να επιβλέπει την διαδικασία ανάπτυξης σε προκαθορισμένα χρονικά διαστήματα, με σκοπό να επιβεβαιώσει ότι η διαδικασία συμμορφώνεται με το συμβόλαιο και είναι σε συνοχή με τα συμφωνηθέντα σχέδια. Οποιαδήποτε αναβάθμιση ή αλλαγή στα σχέδια θα πρέπει να υπόκειται στην έγκριση του ιδιοκτήτη του λογισμικού.(Department Of Defense, 1994)

- 2) Δημιουργία περιβάλλοντος ανάπτυξης λογισμικού. Ο προγραμματιστής θα πρέπει να δημιουργήσει ένα περιβάλλον ανάπτυξης του λογισμικού, σύμφωνα με τις ακόλουθες απαιτήσεις:
- a) Περιβάλλον software engineering. Ο προγραμματιστής θα πρέπει να δημιουργήσει, να ελέγχει και να συντηρεί ένα περιβάλλον software engineering έτσι ώστε να εκτελεστεί όλη τη διαδικασία του software engineering. Θα πρέπει να επιβεβαιώσει ότι κάθε στοιχείο του περιβάλλοντος εκτελεί τις σχεδιασμένες λειτουργίες του,
 - b) Περιβάλλον δοκιμών του λογισμικού. Ο προγραμματιστής πρέπει δημιουργήσει, να ελέγχει και να συντηρεί ένα περιβάλλον δοκιμής του λογισμικού έτσι ώστε να εκτελέσει την πιστοποίηση και άλλες πιθανές δοκιμές του λογισμικού. Θα πρέπει να επιβεβαιώσει ότι κάθε στοιχείο του περιβάλλοντος εκτελεί τις σχεδιασμένες λειτουργίες του.
 - c) Βιβλιοθήκη ανάπτυξης λογισμικού. Ο προγραμματιστής θα πρέπει να δημιουργήσει, να ελέγχει και να συντηρεί μία βιβλιοθήκη ανάπτυξης λογισμικού, έτσι ώστε να βοηθήσει στην κατά διαστήματα ανάπτυξη και επακόλουθη υποστήριξη του λογισμικού. Η βιβλιοθήκη αυτή πρέπει να είναι αναπόσπαστο τμήμα του περιβάλλοντος δοκιμών και software engineering. Ο προγραμματιστής πρέπει να διατηρήσει τη βιβλιοθήκη καθόλη τη διάρκεια του συμβολαίου.
 - d) Αρχεία ανάπτυξης λογισμικού. Ο προγραμματιστής θα πρέπει δημιουργήσει, να ελέγχει και να διατηρεί αρχεία ανάπτυξης λογισμικού για κάθε τμήμα λογισμικού ή ομάδες τμημάτων λογισμικού. Επίσης θα πρέπει να καταγραφεί όλες οι πληροφορίες που σχετίζονται με την ανάπτυξη του λογισμικού στα αντίστοιχα αρχεία ανάπτυξης λογισμικού και να τα διατηρεί καθόλη τη διάρκεια του συμβολαίου.
 - e) Μη παραδοτέο λογισμικό. Ο προγραμματιστής μπορεί να χρησιμοποιήσει μη παραδοτέο λογισμικό κατά την ανάπτυξη του λογισμικού με την προϋπόθεση ότι η χρήση και υποστήριξη του παραδοτέου λογισμικού δεν εξαρτάται από το μη παραδοτέο λογισμικό που χρησιμοποίησε ο προγραμματιστής. (Department Of Defense, 1994)
- 3) Ανάλυση απαιτήσεων συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στην ανάλυση των απαιτήσεων συστήματος σύμφωνα με τις παρακάτω προδιαγραφές:
- a) Ανάλυση απαιτήσεων του χρήστη. Ο προγραμματιστής θα πρέπει να συμμετάσχει στην ανάλυση των απαιτήσεων του χρήστη, οι οποίες παρέχονται από τον ιδιοκτήτη του λογισμικού έτσι ώστε να γίνουν κατανοητές οι ανάγκες του χρήστη. Τα δεδομένα που θα τον βοηθήσουν στην κατανόηση και ανάλυση των

απαιτήσεων του χρήστη μπορεί να είναι συνεντεύξεις, ανατροφοδότηση από χρήση πρωτότυπων, αναφορές προβλημάτων, ερωτηματολόγια ή άλλα δεδομένα χρηστών.

- b) Γενική ιδέα. Ο προγραμματιστής θα πρέπει να συμμετέχει στη διαδικασία ορισμού και καταγραφής της γενικής ιδέας του συστήματος. Το αποτέλεσμα πρέπει να περιλαμβάνει όλα τα πεδία στην “Περιγραφή της γενικής ιδέας”.
 - c) Απαιτήσεις συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στη διαδικασία ορισμού και καταγραφής των απαιτήσεων που πρέπει να πληροί το σύστημα και τις μεθόδους που πρέπει να χρησιμοποιηθούν, έτσι ώστε να εξασφαλιστεί ότι κάθε απαίτηση έχει ικανοποιηθεί. Το αποτέλεσμα πρέπει να περιλαμβάνει όλα τα πεδία στην “Προδιαγραφή συστήματος/υποσυστήματος”.(Department Of Defense, 1994)
- 4) Σχεδίαση συστήματος. Ο προγραμματιστής θα πρέπει να συμμετάσχει στη σχεδίαση του συστήματος σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Αποφάσεις σε επίπεδο συστήματος. Ο προγραμματιστής θα πρέπει να συμμετάσχει στη διαδικασία ορισμού και καταγραφής των αποφάσεων σε επίπεδο συστήματος. Το αποτέλεσμα θα πρέπει να περιλαμβάνει όλα τα πεδία που περιγράφονται στην ενότητα “Αποφάσεις σε επίπεδο συστήματος” της “Προδιαγραφής συστήματος/υποσυστήματος”.
 - b) Αρχιτεκτονικό σχέδιο συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στη διαδικασία ορισμού και καταγραφής του αρχιτεκτονικού σχεδίου του συστήματος (αναγνωρίζοντας τα μέρη του συστήματος, τις διεπαφές και μία ιδέα εκτέλεσής τους) καθώς και την ιχνηλασιμότητα μεταξύ των τμημάτων του συστήματος και των απαιτήσεων του συστήματος. Το αποτέλεσμα θα πρέπει να περιλαμβάνει όλα τα πεδία που περιλαμβάνονται στην ενότητα “Αρχιτεκτονικό σχέδιο και ιχνηλασιμότητα” της “Προδιαγραφής του συστήματος/υποσυστήματος”.
 - c) Ανάλυση απαιτήσεων λογισμικού. Ο προγραμματιστής θα πρέπει να ορίσει και να καταγράψει τις απαιτήσεις του λογισμικού και τις μεθόδους που θα χρησιμοποιηθούν, ώστε να εξασφαλιστεί ότι έχουν ικανοποιηθεί όλες οι απαιτήσεις και η ιχνηλασιμότητα μεταξύ των απαιτήσεων.(Department Of Defense, 1994)
- 5) Σχεδίαση λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελέσει τη σχεδίαση του λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Αποφάσεις σε επίπεδο λογισμικού. Ο προγραμματιστής θα πρέπει να ορίσει και να καταγράψει όλες τις αποφάσεις σε επίπεδο λογισμικού. Το αποτέλεσμα θα

πρέπει να περιλαμβάνει όλα τα πεδία στην ενότητα “Σχεδίαση σε επίπεδο λογισμικού” της “Προδιαγραφής της σχεδίασης Λογισμικού”.

- b) Αρχιτεκτονικό σχέδιο λογισμικού. Ο προγραμματιστής θα πρέπει να συμμετέχει στη διαδικασία ορισμού και καταγραφής του αρχιτεκτονικού σχεδίου του λογισμικού (αναγνωρίζοντας τα μέρη του λογισμικού, τις διεπαφές και μία ιδέα εκτέλεσής τους) καθώς και την ιχνηλασιμότητα μεταξύ των τμημάτων του λογισμικού και των απαιτήσεων του λογισμικού. Το αποτέλεσμα θα πρέπει να περιλαμβάνει όλα τα πεδία που περιλαμβάνονται στην ενότητα “Αρχιτεκτονικό σχέδιο και ιχνηλασιμότητα” της “Προδιαγραφής της σχεδίασης Λογισμικού”.
 - c) Λεπτομερής σχεδίαση λογισμικού. Ο προγραμματιστής θα πρέπει να αναπτύξει και να καταγράψει μία περιγραφή για κάθε μονάδα λογισμικού. Το αποτέλεσμα θα πρέπει να περιλαμβάνει όλα τα πεδία στην ενότητα “Λεπτομερής σχεδίαση” της “Προδιαγραφής της σχεδίασης λογισμικού”.(Department Of Defense, 1994)
- 6) Δημιουργία και δοκιμή του λογισμικού. Ο προγραμματιστής πρέπει να διεξάγει εκτέλεση και δοκιμές του λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις.
- a) Δημιουργία λογισμικού. Ο προγραμματιστής πρέπει να αναπτύξει και να καταγράψει λογισμικό για κάθε μονάδα λογισμικού στο συμφωνηθέν σχέδιο. η δράση αυτή θα πρέπει να περιλαμβάνει δημιουργία κώδικα και ορισμούς δεδομένων, χτίσιμο βάσεων δεδομένων, εισαγωγή δεδομένων στις βάσεις και σε άλλα αρχεία δεδομένων καθώς και οποιαδήποτε άλλη δράση απαιτείται με σκοπό την υλοποίηση του σχεδίου. Η χρήση οποιασδήποτε γλώσσας προγραμματισμού η οποία δεν διευκρινίζεται στο συμβόλαιο, προϋποθέτει την έγκριση του ιδιοκτήτη του λογισμικού.
 - b) Προετοιμασία δοκίμων. Ο προγραμματιστής θα πρέπει να δημιουργήσει σενάρια δοκίμων (εισαγωγή δεδομένων, αναμενόμενα αποτελέσματα και κριτήρια αξιολόγησης), διαδικασίες δοκίμων και δεδομένα για την δοκιμή του λογισμικού. Τα σενάρια δοκίμων θα πρέπει να καλύπτουν όλες τις περιπτώσεις του λεπτομερούς σχεδίου του λογισμικού.
 - c) Εκτέλεση δοκίμων. Ο προγραμματιστής θα πρέπει να δοκιμάσει το λογισμικό σύμφωνα με τις διαδικασίες και τα σενάρια δοκίμων.
 - d) Αναπροσαρμογή και δοκιμή. Ο προγραμματιστής θα πρέπει να εκτελέσει όλες τις αναγκαίες αναπροσαρμογές στο λογισμικό, να εκτελέσει εκ νέου δοκιμές και να ενημερώσει τα αρχεία ανάπτυξης λογισμικού όπως απαιτείται.
 - e) Ανάλυση και καταγραφή αποτελεσμάτων. Ο προγραμματιστής θα πρέπει να αναλύσει και να καταγράψει τα αποτελέσματα των δοκίμων.(Department Of Defense, 1994)

- 7) Ενσωμάτωση λογισμικού και δοκιμή. Ο προγραμματιστής θα πρέπει να εκτελέσει ενσωμάτωση λογισμικού και εκ νέου δοκιμή του σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Προετοιμασία δοκίμων. Ο προγραμματιστής θα πρέπει να δημιουργήσει σενάρια δοκίμων (εισαγωγή δεδομένων, αναμενόμενα αποτελέσματα και κριτήρια αξιολόγησης), διαδικασίες δοκίμων και δεδομένα για την δοκιμή του λογισμικού. Τα σενάρια δοκίμων θα πρέπει να καλύπτουν όλες τις περιπτώσεις του λεπτομερούς σχεδίου του λογισμικού.
 - b) Εκτέλεση δοκίμων. Ο προγραμματιστής θα πρέπει να δοκιμάσει το λογισμικό σύμφωνα με τις διαδικασίες και τα σενάρια δοκίμων.
 - c) Αναπροσαρμογή και δοκιμή. Ο προγραμματιστής θα πρέπει να εκτελέσει όλες τις αναγκαίες αναπροσαρμογές στο λογισμικό, να εκτελέσει εκ νέου δοκιμές και να ενημερώσει τα αρχεία ανάπτυξης λογισμικού όπως απαιτείται.
 - d) Ανάλυση και καταγραφή αποτελεσμάτων. Ο προγραμματιστής θα πρέπει να αναλύσει και να καταγράψει τα αποτελέσματα των δοκίμων.(Department Of Defense, 1994)
- 8) Δοκιμές πιστοποίησης λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελέσει δοκιμές πιστοποίησης λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Ανεξαρτησία δοκίμων πιστοποίησης. Το άτομο (άτομα) που θα εκτελέσουν τις δοκιμές πιστοποίησης, δεν θα πρέπει να είναι τα άτομα που σχεδίασαν ή δημιούργησαν το λογισμικό. Αυτό δεν αποκλείει τους τελευταίους από το να συνεισφέρουν στη διαδικασία των δοκίμων.
 - b) Δοκιμές στο προοριζόμενο υπολογιστικό σύστημα. Οι δοκιμές πιστοποίησης του λογισμικού θα πρέπει να περιλαμβάνουν απευθείας δοκιμές στο προοριζόμενο υπολογιστικό σύστημα ή σε αντίστοιχο κατόπιν έγκρισης του ιδιοκτήτη του λογισμικού.
 - c) Προετοιμασία δοκίμων πιστοποίησης. Ο προγραμματιστής θα πρέπει να ορίσει και να καταγράψει τις προετοιμασίες, τα σενάρια δοκίμων και τις διαδικασίες δοκίμων, οι οποίες θα χρησιμοποιηθούν στις δοκιμές πιστοποίησης. Επίσης θα πρέπει να ετοιμάσει τα δεδομένα δοκίμων, τα οποία είναι απαραίτητα για να εκτελεστούν τα σενάρια και να παρέχει στον ιδιοκτήτη του λογισμικού πληροφορίες για το χρόνο και τον τρόπο της εκτέλεσης των δοκίμων πιστοποίησης.
 - d) Πρόβα δοκίμων πιστοποίησης. Στην περίπτωση που οι δοκιμές πιστοποίησης του λογισμικού γίνουν παρουσία του ιδιοκτήτη του λογισμικού, ο προγραμματιστής θα πρέπει να εκτελέσει μία πρόβα των σεναρίων και των διαδικασιών που θα

χρησιμοποιηθούν στις δοκιμές πιστοποίησης, έτσι ώστε να εξασφαλίσει ότι είναι πλήρεις και ακριβείς καθώς και ότι το λογισμικό είναι έτοιμο για την τελική δοκιμή. Στην περίπτωση αυτή ο προγραμματιστής θα πρέπει να καταγράψει τα αποτελέσματα αυτής της διαδικασίας στα αντίστοιχα αρχεία ανάπτυξης λογισμικού.

- e) Εκτέλεση δοκιμών πιστοποίησης. Ο προγραμματιστής θα πρέπει να εκτελέσει τις δοκιμές πιστοποίησης σύμφωνα με τα σενάρια και τις διαδικασίες δοκιμών.
 - f) Αναπροσαρμογή και δοκιμή. Ο προγραμματιστής θα πρέπει να εκτελέσει τις αναγκαίες αναπροσαρμογές το λογισμικό, να ενημερώσει τον ιδιοκτήτη του λογισμικού για την επικείμενη δοκιμή, να εκτελέσει την δοκιμή και να ενημερώσει τα αντίστοιχα αρχεία ανάπτυξης λογισμικού και άλλα προϊόντα λογισμικού όπως απαιτείται, βασιζόμενος στα αποτελέσματα των δοκιμών πιστοποίησης.
 - g) Ανάλυση και καταγραφή των αποτελεσμάτων των δοκιμών πιστοποίησης. Ο προγραμματιστής θα πρέπει να αναλύσει και να καταγράψει τα αποτελέσματα των δοκιμών πιστοποίησης. Τα αποτελέσματα θα πρέπει να περιλαμβάνουν όλα τα πεδία στην “Αναφορά δοκιμής λογισμικού”.(Department Of Defense, 1994)
- 9) Ολοκλήρωση λογισμικού/hardware και δοκιμή. Ο Προγραμματιστής θα πρέπει να συμμετάσχει στην ολοκλήρωση και δοκιμή λογισμικού / hardware σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Προετοιμασία για ολοκλήρωση λογισμικού / Hardware και δοκιμή. Ο προγραμματιστής θα πρέπει να συμμετέχει στην ανάπτυξη και στην καταγραφή σεναρίων, διαδικασιών και δεδομένων δοκιμών με σκοπό την υλοποίηση της ολοκλήρωσης λογισμικού/Hardware και της δοκιμής της.
 - b) Εκτέλεση της ολοκλήρωσης λογισμικού / hardware και της δοκιμής. Ο προγραμματιστής θα πρέπει να συμμετέχει στην διαδικασία ολοκλήρωσης και δοκιμής λογισμικού / hardware. Οι δοκιμές θα πρέπει να γίνουν σύμφωνα με τα σενάρια και τις διαδικασίες που έχουν καθοριστεί.
 - c) Αναπροσαρμογή και δοκιμή. Ο προγραμματιστής θα πρέπει να κάνει όλες τις απαραίτητες αναπροσαρμογές στο λογισμικό, να συμμετάσχει σε όλες τις απαραίτητες δοκιμές και να ενημερώσει τα αντίστοιχα αρχεία ανάπτυξης λογισμικού και τα άλλα προϊόντα λογισμικού όπως απαιτείται, βασιζόμενος στα αποτελέσματα της ολοκλήρωσης και δοκιμής λογισμικού / hardware.
 - d) Ανάλυση και καταγραφή των αποτελεσμάτων της δοκιμής και ολοκλήρωσης λογισμικού / Hardware. Ο προγραμματιστής θα πρέπει να συμμετάσχει στην ανάλυση των αποτελεσμάτων της ολοκλήρωσης λογισμικού / Hardware και της

δοκιμής της. Η ανάλυση και τα αποτελέσματα που είναι σχετικά με το λογισμικό, θα πρέπει να καταγραφούν στα αντίστοιχα αρχεία ανάπτυξης λογισμικού.(Department Of Defense, 1994)

- 10) Δοκιμή πιστοποίησης συστήματος. Ο προγραμματιστής θα πρέπει να συμμετάσχει στη διαδικασία δοκιμής πιστοποίησης του συστήματος Σύμφωνα με τις ακόλουθες απαιτήσεις:
- a) Ανεξαρτησία δοκιμών πιστοποίησης συστήματος. Τα άτομα τα οποία είναι υπεύθυνα για την εκπλήρωση των απαιτήσεων που περιγράφονται σε αυτή την ενότητα, δεν πρέπει να είναι τα άτομα τα οποία σχεδίασαν λεπτομερώς ή δημιούργησαν το λογισμικό μέσα στο σύστημα. Αυτό δεν αποκλείει τους τελευταίους από το να συνεισφέρουν στην διαδικασία.
 - b) Δοκιμές στο προοριζόμενο υπολογιστικό σύστημα. Οι δοκιμές πιστοποίησης του συστήματος θα πρέπει να περιλαμβάνουν δοκιμές κατευθείαν στο προοριζόμενο υπολογιστικό σύστημα ή σε αντίστοιχο κατόπιν έγκρισης του ιδιοκτήτη του λογισμικού.
 - c) Προετοιμασία δοκιμών πιστοποίησης συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στην ανάπτυξη και την καταγραφή των προετοιμασιών δοκιμής, των σεναρίων και των διαδικασιών δοκιμής, τα οποία θα χρησιμοποιηθούν για την δοκιμή πιστοποίησης του συστήματος, καθώς και την ιχνηλασιμότητα μεταξύ των σεναρίων και των απαιτήσεων του συστήματος. Στην περίπτωση συστημάτων λογισμικού τα αποτελέσματα θα πρέπει να περιλαμβάνουν όλα τα πεδία στην “Προδιαγραφή δοκιμής λογισμικού”. Ο προγραμματιστής θα πρέπει να συμμετέχει στην προετοιμασία των δεδομένων της δοκιμής που είναι απαραίτητα για να εκτελεστούν τα σενάρια δοκιμών και να ενημερώσει για τον χρόνο και τον τόπο που θα εκτελεστεί η δοκιμή πιστοποίησης του συστήματος.
 - d) Πρόβα δοκιμών πιστοποίησης. Στην περίπτωση που οι δοκιμές πιστοποίησης του λογισμικού γίνουν παρουσία του ιδιοκτήτη του λογισμικού, ο προγραμματιστής θα πρέπει να συμμετάσχει στις πρόβες δοκιμής των σεναρίων και των διαδικασιών, με σκοπό να διασφαλιστεί ότι είναι πλήρεις και ακριβείς και ότι το σύστημα είναι έτοιμο για την τελική δοκιμή. Ο προγραμματιστής πρέπει να καταγράψει όλα τα αποτελέσματα σχετιζόμενα με το λογισμικό και να ενημερώσει τα αντίστοιχα αρχεία ανάπτυξης λογισμικού.
 - e) Εκτέλεση δοκιμών πιστοποίησης συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στην δοκιμή πιστοποίησης του συστήματος. Η συμμετοχή θα πρέπει να γίνει σύμφωνα με τα σενάρια και τις διαδικασίες των δοκιμών του συστήματος.

- f) Αναπροσαρμογή και δοκιμή. Ο προγραμματιστής θα πρέπει να εκτελέσει τις αναγκαίες αναπροσαρμογές στο λογισμικό, να ενημερώσει τον ιδιοκτήτη του λογισμικού για την επικείμενη δοκιμή, να εκτελέσει την δοκιμή και να ενημερώσει τα αντίστοιχα αρχεία ανάπτυξης λογισμικού και άλλα προϊόντα λογισμικού όπως απαιτείται, βασιζόμενος στα αποτελέσματα των δοκιμών πιστοποίησης συστήματος.
- g) Ανάλυση και καταγραφή των Αποτελεσμάτων των δοκιμών πιστοποίησης συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στην ανάλυση και καταγραφή των αποτελεσμάτων των δοκιμών πιστοποίησης του συστήματος. Στην περίπτωση συστημάτων λογισμικού, τα αποτελέσματα θα πρέπει να περιλαμβάνουν όλα τα πεδία που βρίσκονται στην “Αναφορά δοκιμών λογισμικού”.(Department Of Defense, 1994)
- 11) Προετοιμασία για χρήση του λογισμικού. Ο προγραμματιστής θα πρέπει να προετοιμαστεί για τη χρήση του λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Προετοιμασία του προς εκτέλεση λογισμικού. Ο προγραμματιστής θα πρέπει να προετοιμάσει το εκτελέσιμο λογισμικό για κάθε χρήστη, συμπεριλαμβανομένων οποιονδήποτε batch files, data files ή άλλα αρχεία λογισμικού τα οποία είναι απαραίτητα για την εγκατάσταση και χρήση του λογισμικού στους προοριζόμενους υπολογιστές.
- b) Προετοιμασία περιγραφής των εκδόσεων για τους χρήστες. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει την ακριβή έκδοση του λογισμικού που προορίζεται για κάθε χρήστη.
- c) Προετοιμασία εγχειριδίων χρήσης. Ο προγραμματιστής θα πρέπει να προετοιμάσει εγχειρίδια χρήσης σύμφωνα με τις παρακάτω απαιτήσεις.
- i) Εγχειρίδια χρήσης λογισμικού. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει τις πληροφορίες που χρειάζονται οι χρήστες του λογισμικού.
- ii) Εγχειρίδια εισόδου-εξόδου λογισμικού. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει τις πληροφορίες που χρειάζονται τα άτομα τα οποία εισάγουν και εξάγουν δεδομένα στο λογισμικό, βασιζόμενοι σε άλλα άτομα για τη χρήση του λογισμικού.
- iii) Εγχειρίδιο κέντρου διαχείρισης λογισμικού. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει τις πληροφορίες που χρειάζονται τα άτομα τα οποία χειρίζονται το λογισμικό σε ένα υπολογιστικό κέντρο ή άλλη δικτυακή εγκατάσταση.

iv) Εγχειρίδια χρήσης υπολογιστών. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει τις πληροφορίες που χρειάζονται για την χρήση των υπολογιστών, στους οποίους τρέχει το εν λόγω λογισμικό.(Department Of Defense, 1994)

12) Προετοιμασία για μετάβαση στο νέο λογισμικό. Ο προγραμματιστής θα πρέπει να προετοιμαστεί για τη μετάβαση στο νέο λογισμικό σύμφωνα με τις παρακάτω απαιτήσεις:

- a) Προετοιμασία για εκτέλεση του λογισμικού. Ο προγραμματιστής θα πρέπει να προετοιμάσει το εκτελέσιμο λογισμικό για κάθε χρήστη, συμπεριλαμβανομένων οποιονδήποτε batch files, data files ή άλλων αρχείων λογισμικού τα οποία είναι απαραίτητα για την εγκατάσταση και χρήση του λογισμικού στους προοριζόμενους υπολογιστές.
- b) Προετοιμασία sourcefiles. Ο προγραμματιστής θα πρέπει να προετοιμάσει source files που θα εγκατασταθούν στο κέντρο υποστήριξης, συμπεριλαμβανομένου οποιονδήποτε batch files, command files, datafiles ή άλλων αρχείων τα οποία είναι απαραίτητα για την επαναδημιουργία του εκτελέσιμου λογισμικού.
- c) Προετοιμασία περιγραφής εκδόσεων για το κέντρο υποστήριξης. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει την ακριβή έκδοση του λογισμικού, η οποία παραδόθηκε στο κέντρο υποστήριξης.
- d) Προετοιμασία του τελικού σχεδίου του λογισμικού και των σχετικών πληροφοριών. Ο προγραμματιστής θα πρέπει να ενημερώσει τα σχέδια του λογισμικού έτσι ώστε να ταιριάζουν με το τελικό προϊόν και θα πρέπει να αναγνωρίσει και να καταγράψει τις μεθόδους, οι οποίες θα χρησιμοποιηθούν για την επαλήθευση των αντιγράφων του λογισμικού, τη χρήση των πόρων του hardware και οποιονδήποτε άλλων πληροφοριών οι οποίες χρειάζονται για την υποστήριξη του λογισμικού.
- e) Ενημέρωση του σχεδίου του συστήματος. Ο προγραμματιστής θα πρέπει να συμμετέχει στην ενημέρωση του σχεδίου του συστήματος, έτσι ώστε να ταιριάζει στο τελικό προϊόν.
- f) Προετοιμασία εγχειριδίων υποστήριξης. Ο προγραμματιστής θα πρέπει να προετοιμάσει εγχειρίδια υποστήριξης σύμφωνα με τις παρακάτω απαιτήσεις:
 - i) Εγχειρίδιο προγραμματισμού υπολογιστών. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει πληροφορίες, οι οποίες είναι απαραίτητες για τον προγραμματισμό των υπολογιστών, στους οποίους δημιουργήθηκε ή πρόκειται να τρέξει το λογισμικό.

- ii) Εγχειρίδια υποστήριξης υλικολογισμικού (Firmware). Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να καταγράψει πληροφορίες, οι οποίες είναι απαραίτητες για τον προγραμματισμό και τον επαναπρογραμματισμό οποιονδήποτε συσκευών υλικολογισμικού, στις οποίες το λογισμικό έχει εγκατασταθεί ή πρόκειται να εγκατασταθεί.
- g) Μεταβίβαση στο κέντρο υποστήριξης. Ο προγραμματιστής θα πρέπει να:
 - i) εγκαταστήσει και να ελέγξει το λογισμικό στο κέντρο υποστήριξης, το οποίο περιγράφεται στο συμβόλαιο
 - ii) εκτελέσει επίδειξη στον ιδιοκτήτη του λογισμικού ότι, το παραδοτέο λογισμικό μπορεί να δημιουργηθεί και να συντηρηθεί χρησιμοποιώντας υλικό και λογισμικό, το οποίο είτε είναι διαθέσιμο στην αγορά, είτε διαθέσιμο στον ιδιοκτήτη ή διαθέσιμο από τον προγραμματιστή.
 - iii) παρέχει εκπαίδευση στο κέντρο υποστήριξης, όπως περιγράφεται στο συμβόλαιο.
 - iv) παρέχει οποιαδήποτε άλλη βοήθεια στο κέντρο υποστήριξης, όπως περιγράφεται στο συμβόλαιο.(Department Of Defense, 1994)
- 13) Διαχείριση διαμόρφωσης λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελεί διαχείριση διαμόρφωσης του λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις:
 - a) Αναγνώριση διαμόρφωσης. Ο προγραμματιστής θα πρέπει να συμμετάσχει στην επιλογή τμημάτων του λογισμικού, τα οποία θα υπόκεινται σε έλεγχο διαμόρφωσης. Το σχεδιάγραμμα αναγνώρισης θα πρέπει να ορίζει σε τι βαθμό και ποιες οντότητες του λογισμικού θα μπορούν να υπόκεινται σε έλεγχο διαμόρφωσης. Το σχεδιάγραμμα αυτό θα πρέπει να περιλαμβάνει την έκδοση/επανεκδοση/ ημερομηνία αποδέσμευσης της κάθε οντότητας.
 - b) Έλεγχος διαμόρφωσης. Ο προγραμματιστής θα πρέπει να δημιουργήσει και να εφαρμόσει διαδικασίες, με τις οποίες θα ανατίθενται τα επίπεδα ελέγχου σε κάθε οντότητα, τα άτομα ή ομάδες ατόμων με εξουσιοδότηση να εξουσιοδοτούν αλλαγές και να κάνουν αλλαγές σε κάθε επίπεδο και τα βήματα τα οποία πρέπει να ακολουθούνται για την αίτηση εξουσιοδότησης αλλαγών.
 - c) Συσκευασία, αποθήκευση, χειρισμός και παράδοση. Ο προγραμματιστής θα πρέπει να δημιουργήσει και να εφαρμόσει διαδικασίες για την συσκευασία, αποθήκευση, χειρισμό και παράδοση των προϊόντων λογισμικού. Επιπλέον, θα πρέπει να κρατά αντίγραφα των παραδοτέων προϊόντων λογισμικού για όλη τη διάρκεια του συμβολαίου.(Department Of Defense, 1994)
- 14) Αξιολόγηση προϊόντος λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελεί αξιολόγηση των προϊόντων λογισμικού σύμφωνα με τις ακόλουθες απαιτήσεις:

- a) Αξιολόγηση των εν εξελίξει και τελικών προϊόντων λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελεί αξιολογήσεις των προϊόντων λογισμικού καθόλη τη διάρκεια της ανάπτυξης. Επιπλέον θα πρέπει να εκτελεί μία τελική αξιολόγηση για κάθε παραδοτέο τμήμα λογισμικού πριν από την παράδοση του.
 - b) Αρχεία αξιολόγησης προϊόντων λογισμικού. Ο προγραμματιστής θα πρέπει να προετοιμάζει και να διατηρεί αρχεία για κάθε αξιολόγηση προϊόντος λογισμικού. Τα αρχεία αυτά θα πρέπει να διατηρούνται καθόλη τη διάρκεια του συμβολαίου.
 - c) Ανεξαρτησία στην αξιολόγηση των προϊόντων λογισμικού. Τα άτομα που είναι υπεύθυνα για την αξιολόγηση των προϊόντων λογισμικού δεν πρέπει να είναι τα άτομα τα οποία ανέπτυξαν το προϊόν. Αυτό δεν αποκλείει τους τελευταίους από το να συμμετέχουν στη διαδικασία της αξιολόγησης. (Department Of Defense, 1994)
- 15) Διασφάλιση ποιότητας λογισμικού. Ο προγραμματιστής θα πρέπει να διασφαλίζει την ποιότητα λογισμικού σύμφωνα με τις παρακάτω απαιτήσεις:
- a) Αξιολόγηση διασφάλισης ποιότητας λογισμικού. Ο προγραμματιστής θα πρέπει να εκτελεί διαρκείς αξιολογήσεις των δραστηριοτήτων ανάπτυξης λογισμικού και των τελικών προϊόντων λογισμικού έτσι ώστε:
 - i) να διασφαλίσει ότι κάθε δραστηριότητα που προβλέπεται από το συμβόλαιο ή περιγράφεται στο σχέδιο ανάπτυξης του λογισμικού, εκτελείται σύμφωνα με το συμβόλαιο
 - ii) να διασφαλίσει ότι κάθε προϊόν υπάρχει και έχει υποβληθεί σε αξιολόγηση της λογισμικού δοκιμές και διορθωτικές ενέργειες, όπως καθορίζεται από το παρόν πρότυπο.
 - b) Αρχεία καταγραφής διασφάλισης ποιότητας λογισμικού. Ο προγραμματιστής θα πρέπει να προετοιμάσει και να διατηρεί αρχεία καταγραφής για κάθε δραστηριότητα διασφάλισης ποιότητας λογισμικού. Τα αρχεία αυτά θα πρέπει να διατηρούνται για όλη τη διάρκεια του συμβολαίου.
 - c) Ανεξαρτησία της διασφάλισης ποιότητας λογισμικού. Τα άτομα που είναι υπεύθυνα για την εκτέλεση των αξιολογήσεων διασφάλισης ποιότητας λογισμικού δεν θα πρέπει να είναι τα άτομα τα οποία ανέπτυξαν το προϊόν λογισμικού ή ενεπλάκησαν σε οποιαδήποτε άλλη δραστηριότητα σχετικά με αυτό. Αυτό δεν αποκλείει τους τελευταίους από το να συμμετάσχουν στη διαδικασία των αξιολογήσεων αυτών. (Department Of Defense, 1994)
- 16) Διορθωτικές ενέργειες. Ο προγραμματιστής θα πρέπει να εκτελεί διορθωτικές ενέργειες σύμφωνα με τις παρακάτω απαιτήσεις.

- a) Αναφορές προβλημάτων / αλλαγών. Ο προγραμματιστής θα πρέπει να προετοιμάζει μία αναφορά προβλήματος/αλλαγής για να περιγράψει κάθε πρόβλημα, το οποίο ανιχνεύεται σε κάποιο από τα προϊόντα λογισμικού ή σε διαδικασίες/δραστηριότητες που προβλέπονται από το συμβόλαιο. Στην αναφορά θα πρέπει να περιγράφεται το πρόβλημα, η διορθωτική ενέργεια που απαιτείται και οι ενέργειες που έχουν γίνει έως εκείνη τη στιγμή. Οι αναφορές αυτές χρησιμεύουν ως δεδομένα εισόδου στο “Σύστημα διορθωτικών ενεργειών”.
- b) Σύστημα διορθωτικών ενεργειών. Ο προγραμματιστής θα πρέπει να εφαρμόσει ένα σύστημα διορθωτικών ενεργειών με σκοπό να διαχειρίζεται κάθε πρόβλημα που ανιχνεύεται σε προϊόντα λογισμικού ή σε διαδικασίες / δραστηριότητες που προβλέπονται από το συμβόλαιο. Το σύστημα πρέπει να ικανοποιεί τις παρακάτω απαιτήσεις:
- i) τα δεδομένα εισόδου του συστήματος θα πρέπει να είναι οι αναφορές προβλημάτων/αλλαγών.
 - ii) το σύστημα πρέπει να είναι κλειστό, έτσι ώστε να διασφαλίζεται ότι όλα τα προβλήματα αναφέρονται αμέσως, εισάγονται στο σύστημα, εκτελούνται άμεσα διορθωτικές ενέργειες, επιτυγχάνεται επίλυση, παρακολουθείται η κατάσταση και γίνεται καταγραφή για όλη τη διάρκεια του συμβολαίου.
 - iii) κάθε πρόβλημα πρέπει να ταξινομείται ανά κατηγορία και προτεραιότητα.
 - iv) θα πρέπει να πραγματοποιείται ανάλυση των προβλημάτων, έτσι ώστε να εντοπίζονται δυσμενείς τάσεις.
 - v) οι διορθωτικές ενέργειες θα πρέπει να αξιολογούνται, έτσι ώστε να καθορίζεται το αν τα προβλήματα επιλύθηκαν, αν δυσμενείς τάσεις αντιστράφηκαν και αν οι αλλαγές εφαρμόστηκαν επιτυχώς χωρίς τη δημιουργία επιπλέον προβλημάτων.(Department Of Defense, 1994)
- 17) Άλλες δραστηριότητες. Ο προγραμματιστής θα πρέπει να εκτελεί τις ακόλουθες δραστηριότητες:
- a) Διαχείριση ρίσκου. Ο προγραμματιστής θα πρέπει να εκτελεί διαχείριση ρίσκου καθόλη τη διάρκεια ανάπτυξης του λογισμικού. Θα πρέπει να αναγνωρίζει, αναλύει και προτεραιοποιεί τα σημεία της διαδικασίας ανάπτυξης του λογισμικού, τα οποία περιλαμβάνουν πιθανούς κινδύνους τεχνικής, οικονομικής, ή προγραμματιστικής φύσεως. Θα πρέπει να αναπτύσσει στρατηγικές διαχείρισης αυτών των κινδύνων, να τους καταγράφει και να εφαρμόζει τις στρατηγικές σύμφωνα με το σχέδιο.
- b) Δείκτες διαχείρισης λογισμικού. Ο προγραμματιστής θα πρέπει να χρησιμοποιήσει δείκτες διαχείρισης λογισμικού, έτσι ώστε να διευκολύνει τη

διαχείριση της διαδικασίας ανάπτυξης και να επικοινωνεί την κατάστασή της στον ιδιοκτήτη του λογισμικού. Ο προγραμματιστής θα πρέπει να αναγνωρίσει και να ορίσει ένα αριθμό τέτοιων δεικτών, συμπεριλαμβανομένων των δεδομένων που πρέπει να συλλεχθούν, των μεθόδων ερμηνείας στους και του μηχανισμού αναφορών.

- c) Ασφάλεια και ιδιωτικότητα. Ο προγραμματιστής θα πρέπει να ικανοποιεί όλες τις απαιτήσεις ασφάλειας και ιδιωτικότητας, οι οποίες περιγράφονται στο συμβόλαιο. Οι απαιτήσεις αυτές μπορεί να επηρεάσουν τη διαδικασία ανάπτυξης του λογισμικού, το τελικό προϊόν ή και τα δύο.
- d) Διαχείριση υπεργολάβων. Στην περίπτωση που χρησιμοποιηθούν υπεργολάβοι, ο προγραμματιστής θα πρέπει να περιλαμβάνει στα υπό συμβόλαιο όλες τις απαιτήσεις που είναι απαραίτητες για να διασφαλιστεί ότι τα προϊόντα λογισμικού αναπτύσσονται με βάση τις απαιτήσεις του κύριου συμβολαίου.
- e) Συντονισμός με συνεργάτες. Ο προγραμματιστής θα πρέπει συντονίζεται είμαι συνεργάτες ή ομάδες συνεργατών όπως ορίζεται στο συμβόλαιο.
- f) Βελτιστοποίηση των διαδικασιών του έργου. Ο προγραμματιστής θα πρέπει σε τακτά χρονικά διαστήματα να αξιολογεί τις διαδικασίες οι οποίες χρησιμοποιούνται στο έργο, έτσι ώστε να καθορίζεται η καταλληλότητα και η αποτελεσματικότητά τους. Βασιζόμενος σε αυτές τις αξιολογήσεις, ο προγραμματιστής θα πρέπει να αναγνωρίζει όλες τις απαραίτητες βελτιώσεις στις διαδικασίες, να τις επικοινωνεί στον ιδιοκτήτη του λογισμικού υπό τη μορφή προτεινόμενων ενημερώσεων στο σχέδιο ανάπτυξης λογισμικού και να τις εφαρμόσει μετά από έγκριση.(Department Of Defense, 1994)

Οι παραπάνω προδιαγραφές αποτελούν τις γενικές κατευθύνσεις για την παραγωγή οποιουδήποτε λογισμικού που αφορά σε στρατιωτική χρήση. Από τις προδιαγραφές αυτές απορρέει ένα μεγάλο πλήθος προτύπων, είτε στρατιωτικών είτε πολιτικών (MIL-STD, IEEE/ANSI std, ISO), τα οποία καθορίζουν λεπτομερώς τις διαδικασίες ανάπτυξης του λογισμικού. Η ανάλυση όλων αυτών των προδιαγραφών ξεπερνά τη φύση της μεταπτυχιακής αυτής εργασίας.

Ως χρήστες παρόμοιων λογισμικών στον εργασιακό μας χώρο, μπορούμε να ορίσουμε τις μη λειτουργικές και λειτουργικές απαιτήσεις που πρέπει να ικανοποιεί η εφαρμογή. Ξεκινώντας από τις μη λειτουργικές απαιτήσεις το πρόγραμμα θα πρέπει:

1. να λειτουργεί αδιάλειπτα για τουλάχιστον οκτώ συνεχόμενες ώρες.
2. να έχει τη δυνατότητα αποθήκευσης των στοιχείων τουλάχιστον 500 χρηστών και 10.000 αποστολών.

3. να μπορεί να χρησιμοποιείται από τουλάχιστον 60 χρήστες ταυτόχρονα.
4. να λειτουργεί σε περιβάλλον browser.
5. να διασφαλίζεται η ιδιωτικότητα των χρηστών σύμφωνα με τα πρότυπα ασφαλείας της Πολεμικής Αεροπορίας.
6. να διασφαλίζεται η ασφάλεια των δεδομένων των αποστόλων σύμφωνα με τα πρότυπα ασφαλείας της Πολεμικής Αεροπορίας.
7. η εφαρμογή θα πρέπει να μπορεί να λειτουργήσει στο υπάρχον δίκτυο της Πολεμικής Αεροπορίας.
8. μόνο οι εξουσιοδοτημένοι εκπαιδευτές θα μπορούν να παραμετροποιούν την εφαρμογή.
9. θα πρέπει να έχει απλή και κατανοητή διεπαφή με τον χρήστη.
10. να μην καταναλώνει πολλούς πόρους του συστήματος.
11. να υπάρχει δυνατότητα αναβάθμισης και εξέλιξής του.
12. να μπορεί να παραμετροποιηθεί με μεγάλη ευκολία χωρίς να χρειάζεται προσωπικό με ειδικές γνώσεις.
13. πιθανή βλάβη του να μην προκαλεί καταστροφή σε υλικό, τραυματισμό ή απώλεια ανθρώπινης ζωής.
14. να μπορεί να δεχτεί αναβαθμίσεις στο μέλλον.

Οι λειτουργικές απαιτήσεις που πρέπει να έχει το πρόγραμμα είναι οι εξής:

1. Το πρόγραμμα θα πρέπει να δέχεται ως δεδομένα, αρχεία .csv.
2. Το πρόγραμμα θα πρέπει να κάνει ανάκτηση ονόματος του μαθητή από το αρχείο τύπου .csv.
3. Το πρόγραμμα θα πρέπει να μπορεί να αξιολογεί όλες τις παραμέτρους που περιλαμβάνονται σε κάθε αρχείο .csv
4. Το πρόγραμμα θα πρέπει να εμφανίζει το όνομα του μαθητή σε εμφανές σημείο πάνω από τον πίνακα αποτελεσμάτων.
5. Το πρόγραμμα θα πρέπει να δέχεται από τον εκπαιδευτή, το εύρος των οποίων θα αξιολογεί τον μαθητή από τον εκπαιδευτή.
6. Το πρόγραμμα θα πρέπει να μπορεί να δέχεται παραμετροποίηση του χρονικού διαστήματος για το οποίο θα αξιολογεί τον μαθητή.
7. Το πρόγραμμα θα πρέπει να αξιολογεί την απόδοση του μαθητή με βάση τα όρια που έχει θέσει ο εκπαιδευτής.

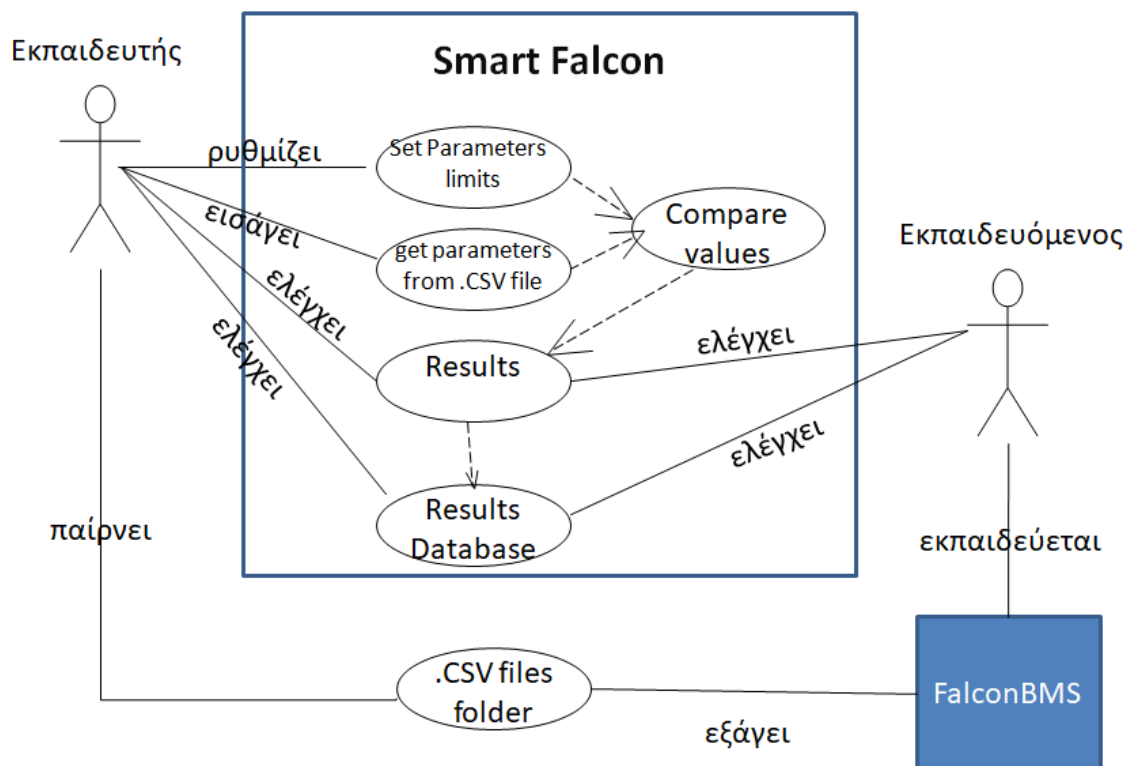
8. Το πρόγραμμα θα πρέπει να αξιολογεί την απόδοση του μαθητή ανά δευτερόλεπτο πτήσης.
9. Το πρόγραμμα θα πρέπει να εμφανίζει τα αποτελέσματα της αξιολόγησης σε μορφή πίνακα στον οποίο να φαίνονται τα παρακάτω στοιχεία:
 - a. Χρόνος.
 - b. Στήλες με τις πραγματικές τιμές των προς αξιολόγηση μεταβλητών όπως ταχύτητα αεροσκάφους, γωνία ανόδου-καθόδου, ύψος, επιτάχυνση της βαρύτητας κ.α. Στο πρόγραμμα που θα δημιουργήσουμε, θα αξιολογούνται μόνο 3 μεταβλητές, ταχύτητα-Speed, πορεία-Heading, επιτάχυνση της βαρύτητας-G-force.
 - c. Θέση του αεροσκάφους με μορφή συντεταγμένων GPS σε δύο στήλες (Latitude και Longitude).
 - d. Στήλες με τα αποτελέσματα των μεταβλητών προς αξιολόγηση (π.χ. EXAM Speed, EXAM Heading, EXAM Altitude κ.α.). Στο πρόγραμμα που θα δημιουργήσουμε, θα αξιολογούνται μόνο 3 μεταβλητές, ταχύτητα-Speed, πορεία-Heading, επιτάχυνση της βαρύτητας-G-force.
 - e. Στήλη στην οποία να φαίνεται το τελικό αποτέλεσμα της αξιολόγησης του μαθητή (Final).
10. Το πρόγραμμα θα πρέπει να εμφανίζει PASS ή FAIL αντίστοιχα στις στήλες στις οποίες αξιολόγησε τον μαθητή.
11. Το πρόγραμμα θα πρέπει να χρωματίζει με πράσινο τα κελιά στα οποία ο μαθητής έχει αξιολογηθεί με PASS και με κόκκινο τα κελιά στα οποία ο μαθητής έχει αξιολογηθεί με FAIL.
12. Το πρόγραμμα θα πρέπει να μπορεί να αξιολογήσει και δεύτερο μαθητή χωρίς να απαιτείται καμία άλλη ενέργεια παρά μόνο να επαναφορτωθεί το αρχείο τύπου .csv από την πτήση του μαθητή.
13. Το πρόγραμμα θα πρέπει να μπορεί να αξιολογήσει τον μαθητή σε οποιαδήποτε παράμετρο του ζητηθεί από τον εκπαιδευτή.
14. Το πρόγραμμα θα πρέπει να δίνει την δυνατότητα στον εξουσιοδοτημένο εκπαιδευτή να έχει πρόσβαση σε όλες τις προηγούμενες πτήσεις του μαθητή.
15. Το πρόγραμμα θα πρέπει να βαθμολογεί τον μαθητή με ποσοστό επί της εκατό ανάλογα με τα πόσα δευτερόλεπτα ο μαθητής ήταν εντός επιθυμητών παραμέτρων στην τελική του απόδοση.
16. Το πρόγραμμα θα πρέπει να δημιουργεί γραφήματα που αφορούν την συνολική απόδοση του μαθητή σε προηγούμενες πτήσεις κατόπιν επιλογής του εκπαιδευτή.

17. Το πρόγραμμα θα πρέπει να δημιουργεί γραφήματα που αφορούν την απόδοση του μαθητή σε προηγούμενες πτήσεις σε πεδία τα οποία θα είναι επιλέξιμα από τον εκπαιδευτή.
18. Το πρόγραμμα θα πρέπει να δίνει την δυνατότητα στον εκπαιδευτή να δημιουργεί σχόλια σχετικά με την απόδοση του μαθητή.

3.3 Διάγραμμα περιπτώσεων χρήσης του *Smart Falcon*

Προκειμένου να περιγράψουμε επιγραμματικά τις περιπτώσεις χρήσης του Smart Falcon, θα χρησιμοποιήσουμε ένα διάγραμμα περιπτώσεων χρήσης. Τα διαγράμματα αυτά χρησιμοποιούνται για να απεικονίσουν μια στατική εικόνα των χρήσεων ενός συστήματος και παίζουν πολύ σημαντικό ρόλο στην οργάνωση και μοντελοποίηση της συμπεριφοράς ενός συστήματος. Ως επί των πλείστων, απεικονίζουν χρήστες και περιπτώσεις χρήσεις, καθώς και τις σχέσεις μεταξύ τους. (Booch, Rumbaugh, & Jacobson, 1998)

Με βάση τις απαιτήσεις που πρέπει να ικανοποιεί το Smart Falcon και εξετάζοντας την απλοποιημένη έκδοχή του προγράμματος, όπως περιγράφηκε στο κεφάλαιο 2, προκύπτει το παρακάτω διάγραμμα (Εικόνα 15).

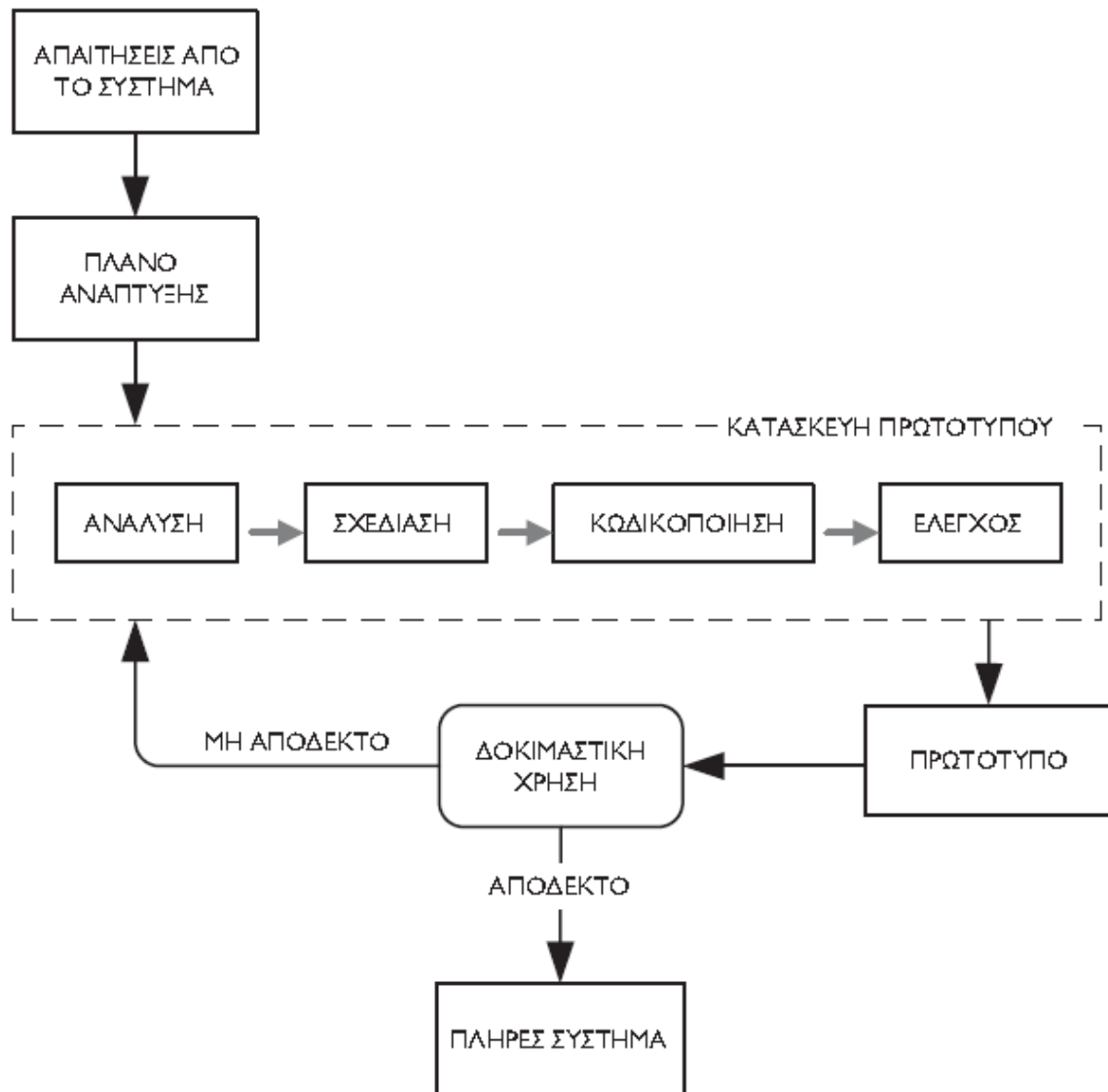


Εικόνα15: Use case diagram Smart Falcon

3.4 Μεθοδολογία ανάπτυξης λογισμικού *Smart Falcon*

Από την ποικιλία των μεθοδολογιών ανάπτυξης λογισμικού, ως καταλληλότερη μέθοδος για την ανάπτυξη του *Smart Falcon* θεωρείται το μοντέλο της προτυποποίησης. Η κεντρική ιδέα του μοντέλου προτυποποίησης (prototyping model) είναι η ανάπτυξη του λογισμικού όχι εξ ολοκλήρου, αλλά σε τμήματα που ονομάζονται «πρωτότυπα». Οι διαδικασίες ανάπτυξης επαναλαμβάνονται για ένα τμήμα του συστήματος κάθε φορά και για το λόγο αυτό το μοντέλο χαρακτηρίζεται ως επαναληπτικό. (Βεσκούκης, 2015) Κάθε πρωτότυπο περιλαμβάνει τις βασικές από τις λειτουργίες που προορίζεται να εκτελεί το λογισμικό και τίθεται σε δοκιμασία από τον πελάτη. Από εκεί συλλέγονται παρατηρήσεις και η διαδικασία κατασκευής νέου πρωτοτύπου επαναλαμβάνεται μέχρις ότου ένα πρωτότυπο να ικανοποιεί τις απαιτήσεις, δηλαδή να εκτελεί τις επιθυμητές λειτουργίες του λογισμικού με τρόπο ικανοποιητικό και να γίνεται αποδεκτό από τον πελάτη (Εικόνα 16). Από το σημείο αυτό και μετά μπορούν να προστεθούν και οι υπόλοιπες λειτουργίες, ώστε το λογισμικό να ολοκληρωθεί. (Βεσκούκης, 2015)

Ένα σημαντικό πλεονέκτημα του μοντέλου αυτού είναι η δυνατότητα απόκτησης άποψης για την εφαρμογή λογισμικού σε σχετικά σύντομο χρονικό διάστημα. Αυτό μπορεί να γλιτώσει την ανάπτυξη από καθυστερήσεις (και συνεπαγόμενα κόσθη) ή ακόμη και από ολική αποτυχία που θα επερχόταν αν ο κατασκευαστής αναγκαζόταν να οπισθοδρομήσει την ανάπτυξη ενώ αυτή είχε προχωρήσει πολύ. (Βεσκούκης, 2015) Παράλληλα, ιδιαίτερη σημασία αποκτά η διοίκηση του έργου η οποία πρέπει να εξασφαλίζει την υλοποιησιμότητα του πρωτοτύπου και την εύκολη τροποποίησή του. Κάθε κατασκευή πρωτοτύπου μπορεί να θεωρηθεί ως ένα μικρό έργο λογισμικού, το οποίο κατασκευάζεται με διαδικασίες που μπορούν να ακολουθούν άλλα μοντέλα κύκλου ζωής, όπως αυτό του καταρράκτη. (Βεσκούκης, 2015) Με βάση τις παραπάνω παρατηρήσεις, το μοντέλο προτυποποίησης χρησιμοποιείται στην ανάπτυξη εφαρμογών λογισμικού για τις απαιτήσεις από τις οποίες δεν υπάρχει βεβαιότητα στην αρχή της ανάπτυξης, οπότε δεν μπορούν να συμφωνηθούν και να παγιωποιηθούν. (Βεσκούκης, 2015) Το *Smart Falcon* ανήκει στην κατηγορία αυτών των εφαρμογών, διότι κατασκευάζεται για πρώτη φορά και είναι στενά εξαρτημένη από την Πολεμική Αεροπορία.



Εικόνα 16: Μέθοδος Πρωτοτυποποίησης(Βεσκούκης, 2015)

4

Δημιουργία προγράμματος

αξιολόγησης Smart Falcon

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο αρχικά θα δούμε τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την δημιουργία του προγράμματος αξιολόγησης καθώς και τους λόγους που αυτές επιλέχθηκαν. Θα εξετάσουμε ορισμένα βασικά κομμάτια του κώδικα που απαρτίζουν το πρόγραμμα και τέλος θα αναλύσουμε τις βασικές λειτουργίες του προγράμματος σε συνδυασμό με κάποια screenshots των επιλογών που μας δίνει.

4.2 Γλώσσες προγραμματισμού που χρησιμοποιήθηκαν

Για την δημιουργία του προγράμματος Smart Falcon, το οποίο θα βοηθάει τον εκπαιδευτή του flight simulator στην αντικειμενική αξιολόγηση του μαθητή-ιπταμένου, χρησιμοποιήθηκαν περισσότερες από μία γλώσσες προγραμματισμού. Συγκεκριμένα χρησιμοποιήθηκε η βιβλιοθήκη React, η οποία χρησιμοποιεί τις γλώσσες προγραμματισμού Javascript, HTML και CSS καθώς και δικά της Tags για την υλοποίηση των προγραμμάτων. Ο λόγος που δεν δημιουργήθηκε ένα πρόγραμμα, το οποίο θα έπρεπε να εγκατασταθεί στο υπολογιστή του χρήστη για να λειτουργήσει είναι ότι θέλουμε να έχουμε την δυνατότητα πρόσβασης στο πρόγραμμα από όλους τους υπολογιστές του τοπικού δικτύου της Πολεμικής Αεροπορίας, χωρίς να υπάρχει η ανάγκη για εγκατάσταση του λογισμικού σε κάθε υπολογιστή ξεχωριστά. Με αυτό τον τρόπο εξασφαλίζουμε ότι οι όλοι οι μαθητές-ιπτάμενοι θα αξιολογούνται πάντα με την τελευταία έκδοση του προγράμματος. Τέλος ικανοποιούμε και το κριτήριο της ασφάλειας από του χρήστες που το έχουν ως απαίτηση (military εφαρμογές), αφού το πρόγραμμά μας θα διασφαλίζεται από τα πρωτόκολλα ασφαλείας του δικτύου της Πολεμικής Αεροπορίας.

4.3 Javascript

Η JavaScript (JS) είναι μια υψηλού επιπέδου διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού. Στην αρχή αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού. Έδωσε τη δυνατότητα με τη χρήση σεναρίων (scripts) από την πλευρά του πελάτη (client-side) να μπορούν οι φυλλομετρητές ιστού να αλληλεπιδρούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα με κάποιο απομακρυσμένο εξυπηρετητή (server) καθώς και να αλλάζουν δυναμικά το περιεχόμενο του HTML (HypertextMarkupLanguage) εγγράφου που εμφανίζεται σε αυτούς. Η Javascript είναι μια γλώσσα σεναρίων (scripted), είναι δυναμική (dynamic) με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης (first-class functions). Η σύνταξή της είναι επηρεασμένη από τη γλώσσα προγραμματισμού C, αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JS προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας τόσο το αντικειμενοστρεφές, το προστακτικό όσο και το συναρτησιακό στυλ προγραμματισμού. Η JS χρησιμοποιείται και σε εφαρμογές, εκτός ιστοσελίδων, όπως έγγραφα PDF, εξειδικευμένοι φυλλομετρητές (site-specific browsers) και μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript, όπως το Node.js, έχουν επίσης κάνει τη Javascript δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript (RussFerguson, 2013).

4.4 React.js

Η βιβλιοθήκη React.js που όπως φαίνεται και από της κατάληξή της αποτελεί βιβλιοθήκη της γλώσσας javascript δημιουργήθηκε από τον JordanWalke, μηχανικό λογισμικού του Facebook. Χρησιμοποιήθηκε για πρώτη φορά στη ροή ενημερώσεων (newsfeed) του Facebook το 2011 και αργότερα στο Instagram το 2012. Έγινε κώδικας ανοιχτού λογισμικού (open-sourced) στο JSConf US το Μάιο του 2013. από το facebook για να λύσει προβλήματα του παρελθόντος. Πιο συγκεκριμένα μια εφαρμογή που χρησιμοποιεί την React.js αποτελείται από αρθρωτά, συνεκτικά και επαναχρησιμοποιήσιμα στοιχεία που συνδυάζονται για τη δημιουργία διεπαφών χρήστη. (Facebook Inc., 2020)

4.5 Smart Falcon User Interface

Σε αυτό το σημείο της εργασίας θα δούμε το UserInterface της εφαρμογής Smart Falcon και θα αναλύσουμε ορισμένα κομμάτια από τον κώδικα της εφαρμογής τα οποία αφορούν το UserInterface της. Η βασική διεπαφή χρήστη εφαρμογής έχει την παρακάτω μορφή (Εικόνα 17).

Step 1:

1 TAS - MAX:0 2 TAS - MIN:0 3 Heading:0 4 gForce:0.1 5 Submit

Step 2:

Click or Drag to Upload

Drop CSV file here or click to upload. 6

Time Start 09:11 PM 8

Time End 09:11 PM 9

Chosen Parameters: 7

TAS MAX	TAS MIN	HEADING MAX	HEADING MIN	G MAX	G MIN
0	0	10	-10	0.2	0

Pilot Exam Table

Pilot Name: 10 11

Time	Heading	Altitude	TAS	gForce	Latitude	Longitude	EXAM TAS	EXAM G	EXAM JHeading	Final
------	---------	----------	-----	--------	----------	-----------	----------	--------	---------------	-------

Εικόνα17: Smart Falcon User Interface

Στα πεδία του Step 1 γίνεται εισαγωγή από τον εκπαιδευτή των παραμέτρων βάσει των οποίων θα εξετάσει στην συνέχεια το πρόγραμμα Smart Falcon την απόδοση του μαθητή. Αναλυτικότερα:

1. Εισαγωγή TAS - MAX. Στο πεδίο αυτό ο εκπαιδευτής εισάγει το μέγιστο όριο της ταχύτητας το οποίο θα αποτελέσει και το όριο από το πρόγραμμα για την αξιολόγηση του μαθητή. Στην συνέχεια φαίνεται ο κώδικας με τον οποίο δημιουργήθηκε το παραπάνω πεδίο (Εικόνα 18).

```

207
208
209
210
211
212
213
214
215
216
217
218
<label>
  TAS - MAX:
  <input
    step="any"
    min="0"
    max="9999999999"
    type="number"
    name="tasValueOriginalMax"
    value={this.state.tasValueOriginalMax}
    onChange={this.handleFiltersChange}
  />{" "}
</label>

```

Εικόνα 17: Κώδικας δημιουργίας πεδίου TAS-MAX

2. Εισαγωγή TAS - MIN. Στο πεδίο αυτό ο εκπαιδευτής εισάγει το ελάχιστο όριο της ταχύτητας το οποίο θα αποτελέσει και το όριο από το πρόγραμμα για την αξιολόγηση του μαθητή.
3. Heading. Στο πεδίο αυτό ο εκπαιδευτής εισάγει την πορεία την οποία θέλει να διατηρήσει ο μαθητής σε όλη την διάρκεια της αξιολόγησης.
4. gForce. Στο πεδίο αυτό ο εκπαιδευτής εισάγει τις δυνάμεις G που θέλει να διατηρεί ο μαθητής για όλη την διάρκεια της αξιολόγησης. Στο απλό παράδειγμα που εξετάζεται στην παρούσα διπλωματική εργασία ο μαθητής αξιολογείται για την ικανότητά του να πετάει το αεροσκάφος σε συνθήκες πτήσης δι' οργάνων και για αυτό τον λόγο το όριο MIN και MAX είναι μόνο $\pm 0,1$.
5. Κουμπί Submit. Με την χρήση αυτού του κουμπιού ο εκπαιδευτής επιβεβαιώνει τις τιμές τις οποίες έθεσε στα προηγούμενα πεδία και ολοκληρώνει την διαδικασία του βήματος 1.

Στα πεδία του Step 2 γίνεται εισαγωγή από τον εκπαιδευτή του CSV file το οποίο περιέχει όλα τα δεδομένα από την πτήση του μαθητή στον εξομοιωτή Falcon BMS. Το αρχείο αυτό το παίρνει ο εκπαιδευτής από το λογισμικό Tacview (πηγή). Τα αρχεία καταγραφής που δημιουργούνται από την πτήση στον εξομοιωτή Falcon BMS είναι αρχεία .vhs. Για τον λόγο αυτό χρησιμοποιούμε το Tacview προκειμένου να μετατραπούν σε .csv αρχεία. Αναλυτικότερα:

6. Πεδίο Upload. Στο πεδίο αυτό ο εκπαιδευτής έχει δύο επιλογές για να εισάγει το αρχείο. Η πρώτη επιλογή είναι κάνοντας αριστερό κλικ πάνω στο πεδίο αυτό. Με την ενέργεια αυτή ανοίγει ο filemanager των Windows για την εισαγωγή του αρχείου. Η δεύτερη επιλογή είναι να γίνει η εισαγωγή του αρχείου με drag and drop. Σε αυτή την περίπτωση ο εκπαιδευτής απλά επιλέγει το αρχείο και στην συνέχεια το τοποθετεί στο πεδίο 6 για να γίνει η εισαγωγή του στο λογισμικό αξιολόγησης. Ο κώδικας με τον οποίο υλοποιείται η παραπάνω διεπαφή φαίνεται παρακάτω στην Εικόνα 19.

```

45 <h5>Click or Drag to Upload</h5>
46 <CSVReader
47   onDrop={this.handleOnDrop}
48   onError={this.handleOnError}
49   config={{
50     header: true,
51     dynamicTyping: true,
52     skipEmptyLines: true,
53   }}
54   addRemoveButton
55   onRemoveFile={this.handleOnRemoveFile}
56 >
57   <span>Drop CSV file here or click to upload.</span>
58 </CSVReader>

```

Εικόνα 18: Κώδικας για την δημιουργία του πεδίου Upload

7. Πίνακας Chosen Parameters. Στο πεδίο αυτό δημιουργείται ένας πίνακας με τις επιλογές τις οποίες έχει κάνει ο χρήστης-εκπαιδευτής του προγράμματος ο οποίος οπτικοποιεί τις επιλογές που έχει κάνει στο Step 1.
8. Πεδίο Time Start. Στο πεδίο αυτό δίνεται η δυνατότητα στον χρήστη-εκπαιδευτή να επιλέξει την ώρα από την οποία θέλει το πρόγραμμα να αξιολογήσει τον μαθητή. Ο λόγος που δίνεται αυτή η δυνατότητα στον χρήστη-εκπαιδευτή είναι για μελλοντική ανάπτυξη του προγράμματος για την αξιολόγηση του μαθητή σε περισσότερες σύνθετες ασκήσεις. Η εισαγωγή της ώρας μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι απλά να γράψει την ώρα ο χρήστης στο πεδίο και ο δεύτερος τρόπος είναι να κάνει αριστερό κλικ πάνω στο ημερολόγιο το οποίο εμφανίζεται και με την βοήθεια ενός γραφήματος (Εικόνα 20) να γίνει η εισαγωγή της ώρας.



Εικόνα 19: Πεδίο Time Start

9. Πεδίο Time End. Στο πεδίο αυτό δίνεται η δυνατότητα στον χρήστη-εκπαιδευτή να επιλέξει την ώρα μέχρι την οποία θέλει το πρόγραμμα να αξιολογήσει τον μαθητή. Η εισαγωγή της ώρας γίνεται ακριβώς με τον τρόπο που αναλύσαμε στο πεδίο 8.
10. Πεδίο Pilot Name. Στο πεδίο αυτό ο χρήστης-εκπαιδευτής δεν χρειάζεται να εκτελέσει καμία ενέργεια αφού το πρόγραμμα κάνει αυτόματη ανάκτηση του ονόματος του εκπαιδευόμενου από το CSV αρχείο που έχει εισαχθεί στο πεδίο 6.
11. Πεδίο Αποτελεσμάτων. Στο πεδίο αυτό δημιουργείται ένας δυναμικός πίνακας με τα αποτελέσματα από την πτήση του μαθητή με βήμα 1 sec. Επίσης δίνονται και τα αποτελέσματα από την αξιολόγηση του μαθητή με μορφή PASS or FAIL και χρωματική απεικόνιση (πράσινο για το PASS και κόκκινο για το FAIL). Παράδειγμα από τον κώδικα που εκτελεί τη παραπάνω διεργασία φαίνεται στην συνέχεια (Εικόνα 21).

```

106         return (
107             <tr key={index}>
108                 <td>
109                     {finalData.data["ISO time"].getHours()}
110                     {" : "}
111                     {finalData.data["ISO time"].getMinutes()}
112                     {" : "}
113                     {finalData.data["ISO time"].getSeconds()}
114                 </td>
115                 <td>{finalData.data.Heading}</td>
116                 <td>{finalData.data.Altitude}</td>
117                 <td>{finalData.data.TAS}</td>
118                 <td>{finalData.data.G}</td>
119                 <td>{finalData.data.Latitude}</td>
120                 <td>{finalData.data.Longitude}</td>
121                 <td>
122                     className={
123                         this.handleIfTasPassOrFail(finalData.data.TAS)
124                         ? "green"
125                         : "red"
126                     }>
127                     {this.handleIfTasPassOrFail(finalData.data.TAS)
128                     ? "PASS"
129                     : "FAIL"}
130                 </td>

```

Εικόνα 20: Κώδικας για την δημιουργία πίνακα αποτελεσμάτων

4.6 Smart Falcon Functions

Για να εκτελέσει την αξιολόγηση του εκπαιδευόμενου το πρόγραμμα χρησιμοποιεί ορισμένες συναρτήσεις. Σε αυτό το σημείο της διπλωματικής εργασίας θα εξετάσουμε τρεις από τις κυριότερες.

4.6.1 *handleIfTasPassOrFail()*

Η συνάρτηση *handleIfTasPassOrFail()* είναι υπεύθυνη να συγκρίνει για κάθε δευτερόλεπτο εάν ο μαθητής κράτησε την ταχύτητα του αεροσκάφους μέσα στα επιθυμητά όρια. Ο κώδικας που πραγματοποιεί αυτό τον έλεγχο παρουσιάζεται στην Εικόνα 22.

```

40     handleIfTasPassOrFail = (originTas) => {
41         if (parseFloat(this.state.tasValueOriginalMax) <= originTas) {
42             return false;
43         } else if (originTas <= parseFloat(this.state.tasValueOriginalMin)) {
44             return false;
45         } else {
46             return true;
47         }
48     };

```

Εικόνα 21: Κώδικας της συνάρτησης handleIfTasPassOrFail()

Παρατηρούμε ότι για να εκτελέσει το έργο του πέρα από τους ελέγχους που εκτελεί συγκρίνοντας την ταχύτητα του μαθητή με τα όρια που έχει θέσει ο εκπαιδευτής καλεί και μία άλλη συνάρτηση την `parseFloat()` με την οποία μετατρέπει τα δεδομένα τύπου `String` σε δεδομένα τύπου `Float`.

4.6.2 *handleIfHeadingPassOrFail()*

Η συνάρτηση `handleIfHeadingPassOrFail()` είναι υπεύθυνη να συγκρίνει σε κάθε δευτερόλεπτο εάν ο μαθητής κράτησε την πορεία του αεροσκάφους μέσα στα επιθυμητά όρια. Ο κώδικας που πραγματοποιεί αυτό τον έλεγχο παρουσιάζεται στην Εικόνα 23.

```

53     handleIfHeadingPassOrFail = (originHeading) => {
54         const originalinputValue = this.state.originHeadingInput;
55         const originHeadingMin = parseFloat(originalinputValue) - 10.0;
56         const originHeadingMax = parseFloat(originalinputValue) + 10.0;
57         if (originHeadingMax <= originHeading) {
58             return false;
59         } else if (originHeading <= originHeadingMin) {
60             return false;
61         } else {
62             return true;
63         }
64     };

```

Εικόνα 22: Κώδικας της συνάρτησης handleIfHeadingPassOrFail()

Παρατηρούμε ότι και σε αυτή την συνάρτηση πέρα από τις συγκρίσεις με την βοήθεια της δομής `if().... elseif()` για τον ίδιο λόγο όπως και στην `handleIfTasPassOrFail()` καλείται η συνάρτηση `parseFloat()`.

4.6.3 *handleIfgForcePassOrFail()*

Η συνάρτηση *handleIfHeadingPassOrFail()* είναι υπεύθυνη να συγκρίνει για κάθε δευτερόλεπτο εάν ο μαθητής κράτησε τα “G” του αεροσκάφους μέσα στα επιθυμητά όρια. Ο κώδικας που πραγματοποιεί αυτό τον έλεγχο παρουσιάζεται στην Εικόνα 24.

```

70     handleIfgForcePassOrFail = (origingForce) => {
71         const originalGForceinputValue = this.state.origingForceInput;
72         const gForceValueMin = parseFloat(originalGForceinputValue) - 0.1;
73         const gForceValueMax = parseFloat(originalGForceinputValue) + 0.1;
74         if (gForceValueMax <= origingForce) {
75             return false;
76         } else if (origingForce <= gForceValueMin) {
77             return false;
78         } else {
79             return true;
80         }
81     };
82

```

Εικόνα 23: Κώδικας της συνάρτησης *handleIfgForcePassOrFail()*

Παρατηρούμε ότι και σε αυτή την συνάρτηση πέρα από τις συγκρίσεις με την βοήθεια της δομής *if().... elseif()* για τον ίδιο λόγο όπως και στην *handleIfTasPassOrFail()* καλείται η συνάρτηση *parseFloat()*.

5

Δοκιμές λογισμικού Smart Falcon

5.1 Εισαγωγή

Ένα από τα πιο σημαντικά μέρη του κύκλου ανάπτυξης λογισμικού είναι η δοκιμή του λογισμικού. Στο κεφάλαιο αυτό θα επικεντρωθούμε στις δοκιμές του λογισμικού Smart Falcon, σε συνδυασμό με όλες τις εργασίες που πρέπει να γίνουν από τον χρήστη του λογισμικού αυτού για να παραχθούν τα τελικά δεδομένα. Κάθε υποκεφάλαιο που ακολουθεί αποτελεί και ένα βήμα που θα πρέπει να εκτελέσει ο χρήστης του λογισμικού για να φτάσει στα αποτελέσματα της αξιολόγησης.

5.2 Βήμα 1ο: Πτήση στον εξομοιωτή με το λογισμικό Falcon BMS 4.32

Το λογισμικό Falcon BMS 4.32 είναι ένα λογισμικό εξομοίωσης πτήσης με F-16 Block 50/52. Είναι λογισμικό ανοικτού κώδικα και βασίζεται στο λογισμικό του εξομοιωτή Falcon 4.0. Η ιστορία αυτού του εξομοιωτή ξεκινά από το 1998 όταν η εταιρεία Microprose τον έβγαλε σε κυκλοφορία και διαθέσιμο στο ευρύ κοινό. Δεν ήταν όμως μόνο ένας εξομοιωτής πτήσης αλλά και ένας εξομοιωτής στρατηγικής. Τον Μάιο του 2004 κυκλοφόρησε η πρώτη έκδοση του εξομοιωτή πτήσης Falcon BMS ο εξομοιωτής αυτός δανείζεται χαρακτηριστικά από τον εξομοιωτή Falcon 4.0 όμως διαθέτει αναβαθμισμένο γραφικό περιβάλλον προσανατολισμένο προς εξομοίωση της πραγματικότητας στον μέγιστο δυνατό βαθμό (Εικόνα 25).



Εικόνα 24: Ρεαλιστικότητα Falcon BMS

Ο ρεαλισμός του Falcon BMS φτάνει σε βαθμό που στις διαδικασίες τις οποίες πρέπει να εκτελέσει ο εκπαιδευόμενος περιλαμβάνεται ακόμη και το άνοιγμα του διακόπτη AVTR (Airborne Video Tape Recorder) πράγμα που πρέπει να κάνει και σε μια πραγματική πτήση. Συνοψίζοντας το πρώτο βήμα περιλαμβάνει την πτήση στον εξομοιωτή με το AVTR σε θέση ON (Εικόνα 26).

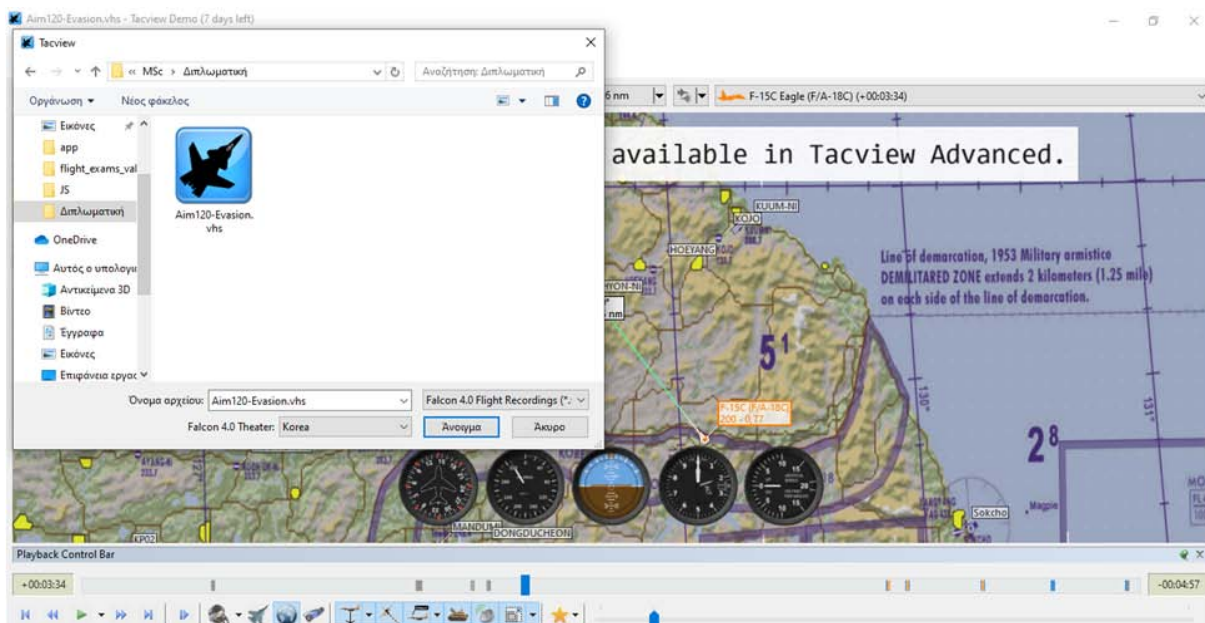


Εικόνα25: AVTR F-16 (Airborne Video Tape Recorder)

5.3 Βήμα 2ο: Χρήση του λογισμικού Tacview

Ο εξομοιωτής Falcon BMS με την χρήση του διακόπτη AVTR σε θέση ON δημιουργεί ένα αρχείο της μορφής example.vhs. Μέσα σε αυτό το αρχείο υπάρχουν καταγεγραμμένα όλα τα δεδομένα του αεροσκάφους με μεγάλη λεπτομέρεια. Επιπρόσθετα σε αυτό το αρχείο καταγράφονται δεδομένα που αφορούν όλα τα αεροσκάφη που πετούν στην εξομοίωση. Το πρόβλημα με αυτό αρχείο είναι ότι δεν είναι σε μορφή που μπορούμε να το επεξεργαστούμε. Για να ξεπεράσουμε αυτό το πρόβλημα χρησιμοποιούμε το λογισμικό Tacview (λογισμικό το οποίο ειδικεύεται στην εκτέλεση απενημέρωσης πτήσης) μέσω του οποίου θα πάρουμε ένα αρχείο CSV (example.csv) το οποίο και μπορεί να επεξεργαστεί το λογισμικό Smart Falcon για να αξιολογήσει τον εκπαιδευόμενο. Ο τρόπος με τον οποίο θα πάρουμε το αρχείο example.csv περιγράφεται στην συνέχεια:

- 1) Ανοίγουμε το λογισμικό Tacview.
- 2) Ανοίγουμε το αρχείο example.vhs από το λογισμικό Tacview (Εικόνα 27).



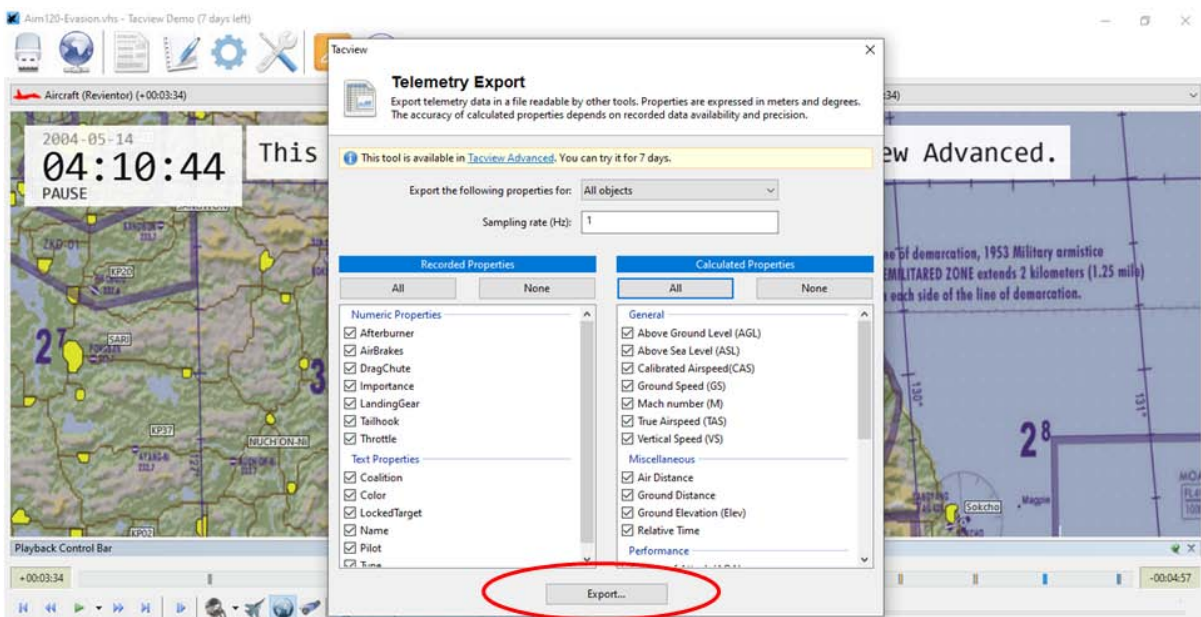
Εικόνα 26: Άνοιγμα αρχείου example.vhs με το Tacview

- 3) Ενεργοποιούμε την λειτουργία Export Telemetry στο Tacview (Εικόνα 28).



Εικόνα 27: Λειτουργία Export Telemetry στο Tacview

- 4) Από τον πίνακα ελέγχου που εμφανίζεται επιλέγουμε τα στοιχεία τα οποία χρειαζόμαστε για την αξιολόγηση και επιλέγουμε το Export (Εικόνα 29).



Εικόνα 28: Ολοκλήρωση διαδικασίας Export Telemetry στο Tacview

Μετά από αυτές τις ενέργειες έχουμε δημιουργήσει το αρχείο example.csv και είναι έτοιμο για χρήση από το πρόγραμμά μας.

5.4 Βήμα 3ο. Χρήση του λογισμικού *Smart Falcon*

Σε αυτό το σημείο ο εκπαιδευτής πρέπει να εκτελέσει 2 απλά βήματα για μπορέσει να έχει από το λογισμικό τα αποτελέσματα της αξιολόγησης του μαθητή. Το πρώτο (Step 1) είναι να ορίσει τα όρια, τα οποία θέλει το λογισμικό να εξετάσει την απόδοση του μαθητή και το δεύτερο (Step 2) είναι να εισάγει το CSV αρχείο του μαθητή στο πρόγραμμα (Εικόνα 30). Κάνοντας και αυτά τα 2 πολύ απλά βήματα ο εκπαιδευτής θα έχει αμέσως τα αποτελέσματα της αξιολόγησης του μαθητή από το λογισμικό, πράγμα το οποίο θα δείξει στον εκπαιδευτή εάν ο μαθητής είναι έτοιμος να προχωρήσει σε περισσότερο προχωρημένες ασκήσεις. Ο εκτιμώμενος χρόνος για να εκτελεστούν αυτά τα βήματα είναι μόλις 1 λεπτό.

Step 1:

TAS - MAX:350 TAS - MIN:150 Heading:135 gForce:1

Step 2:

Click or Drag to Upload

Drop CSV file here or click to upload.

Time Start
09:11 PM

Time End
09:11 PM

Chosen Parameters:

TAS	TAS	HEADING	HEADING	G	G
MAX	MIN	MAX	MIN	MAX	MIN
350	150	145	125	1.1	0.9

Pilot Exam Table

Pilot Name:

Time	Heading	Altitude	TAS	gForce	Latitude	Longitude	EXAM TAS	EXAM G	EXAM JHeading	Final
------	---------	----------	-----	--------	----------	-----------	----------	--------	---------------	-------

Εικόνα29: Step 1 και Step 2 στο Smart Falcon

5.5 Αποτελέσματα Χρήσης

Για να έχουμε μια ολοκληρωμένη εικόνα της λειτουργίας του προγράμματος Smart Falcon θα δούμε δύο περιπτώσεις. Στην πρώτη περίπτωση ο εκπαιδευόμενος θα κινείται εντός των επιθυμητών ορίων (Περίπτωση Α) και θα περατώνει την εκπαίδευση ενώ στην δεύτερη περίπτωση (Περίπτωση Β) θα αποτυγχάνει και θα χρειάζεται επιπρόσθετη εκπαίδευση.

5.5.1 Περίπτωση Α

Σε αυτή την περίπτωση ο εκπαιδευτής θέλει να εξετάσει εάν ο μαθητής διατήρησε μέγιστη ταχύτητα 280 Knots, ελάχιστη ταχύτητα 320 Knots, πορεία 135° με

απόκλιση $\pm 10^\circ$ και τα G του αεροσκάφους να είναι από 0,9 μέχρι 1,1. Μόλις ο εκπαιδευτής εισάγει τα δεδομένα στο πρόγραμμα αυτό θα δημιουργήσει τον παρακάτω πίνακα (Εικόνα 31) και θα δώσει μετά από επεξεργασία των στοιχείων τα παρακάτω αποτελέσματα (Εικόνα 32).

Chosen Parameters:

TAS MAX	TAS MIN	HEADING MAX	HEADING MIN	G MAX	G MIN
320	280	145	125	1.1	0.9

Εικόνα 30: Προς εξέταση παράμετροι (Περίπτωση Α)

Pilot Exam Table

Pilot Name: Revientor

Time	Heading	Altitude	TAS	gForce	Latitude	Longitude	EXAM TAS	EXAM G	EXAM JHeading	Final
7:11:41	135.6	5856.95	282.971723242421	0.993285934217823	38.884523	127.2215467	PASS	PASS	PASS	PASS
7:11:42	135.7	5861.75	285.426706424493	0.998939415895881	38.8826999	127.2233645	PASS	PASS	PASS	PASS
7:11:43	135.8	5866.58	286.247233016788	0.997114899237512	38.8808683	127.2251842	PASS	PASS	PASS	PASS
7:11:44	135.9	5871.46	287.665531929941	0.995742205214011	38.8790247	127.2270097	PASS	PASS	PASS	PASS
7:11:45	136	5876.38	289.327968855501	0.996808917849373	38.877167	127.2288423	PASS	PASS	PASS	PASS
7:11:46	136.1	5881.36	291.035344776522	1.00823646209782	38.8752954	127.2306824	PASS	PASS	PASS	PASS
7:11:47	136	5886.33	289.183710918497	0.992562044001277	38.8734324	127.2325084	PASS	PASS	PASS	PASS
7:11:48	136	5891.44	293.317651914752	1.00593429322035	38.8715439	127.2343609	PASS	PASS	PASS	PASS
7:11:49	135.9	5896.55	292.010821195847	0.985329323409284	38.8696654	127.2362065	PASS	PASS	PASS	PASS
7:11:50	135.9	5901.76	296.804376793041	1.00884182459044	38.8677577	127.2380843	PASS	PASS	PASS	PASS
7:11:51	135.8	5906.93	293.796193027177	0.993223902216156	38.8658706	127.2399446	PASS	PASS	PASS	PASS
7:11:52	135.8	5912.14	296.180640369213	0.99387422064419	38.8639701	127.2418216	PASS	PASS	PASS	PASS
7:11:53	135.7	5917.38	297.902517133875	1.00240372563938	38.8620599	127.2437115	PASS	PASS	PASS	PASS
7:11:54	135.6	5922.63	297.281243332699	1.00401540828624	38.8601563	127.2455994	PASS	PASS	PASS	PASS
7:11:55	135.4	5927.85	296.305242767003	0.982885254062714	38.8582624	127.247485	PASS	PASS	PASS	PASS
7:11:56	135.3	5933.12	301.318501775297	0.993163925363884	38.8563416	127.2494076	PASS	PASS	PASS	PASS
7:11:57	135.1	5938.23	299.436942607796	0.963972515740997	38.8544374	127.2513226	PASS	PASS	PASS	PASS
7:11:58	135.1	5942.9	297.972437460796	0.954196216822838	38.8525462	127.2532328	PASS	PASS	PASS	PASS
7:11:59	135	5947.25	301.062132586698	0.999912874396787	38.8506385	127.2551654	PASS	PASS	PASS	PASS
7:12:0	134.9	5951.5	298.787079288914	0.99088545877334	38.8487481	127.2570864	PASS	PASS	PASS	PASS
7:12:1	135.2	5955.72	300.068421540791	0.994829559824086	38.8468496	127.2590158	PASS	PASS	PASS	PASS
7:12:2	135.3	5959.83	299.289778091701	0.997353215684053	38.8449505	127.2609344	PASS	PASS	PASS	PASS
7:12:3	135.4	5963.88	300.700068561333	0.998310850175603	38.8430474	127.2628483	PASS	PASS	PASS	PASS

Εικόνα 31: Αποτελέσματα αξιολόγησης από το Smart Falcon (Περίπτωση Α)

Όπως βλέπουμε στις παραπάνω εικόνες το πρόγραμμα δείχνει ότι ο μαθητής περάτωσε την εκπαίδευση αφού σε πολλές φάσεις της πτήσης κατάφερε να κινείται εντός των επιθυμητών παραμέτρων (στήλη Final του πίνακα).

5.5.2 Περίπτωση Β

Σε αυτή την περίπτωση ο εκπαιδευτής θέλει να εξετάσει εάν ο μαθητής διατήρησε μέγιστη ταχύτητα 350 Knots, ελάχιστη ταχύτητα 150 Knots, πορεία 135° με

απόκλιση $\pm 10^\circ$ και τα G του αεροσκάφους να είναι από 0,9 μέχρι 1,1. Μόλις ο εκπαιδευτής εισάγει τα δεδομένα στο πρόγραμμα αυτό θα δημιουργήσει τον παρακάτω πίνακα (Εικόνα 33) και θα δώσει μετά από επεξεργασία των στο χείρων τα παρακάτω αποτελέσματα (Εικόνα 34).

Chosen Parameters:

TAS MAX	TAS MIN	HEADING MAX	HEADING MIN	G MAX	G MIN
350	150	145	125	1.1	0.9

Εικόνα 32: Προς εξέταση παράμετροι (Περίπτωση Β)

Pilot Exam Table

Pilot Name: Revientor

Time	Heading	Altitude	TAS	gForce	Latitude	Longitude	EXAM TAS	EXAM G	EXAM JHeading	Final
7:10:45	135.5	6095.46	223.400454812731	0.865846223496786	38.9742617	127.1295472	PASS	FAIL	PASS	FAIL
7:10:46	135.5	6091.07	223.400454812731	1.00661337397461	38.9728411	127.1309763	PASS	FAIL	PASS	FAIL
7:10:47	135.4	6085.42	224.8206851202	1.04608635796183	38.9714118	127.1324137	PASS	PASS	PASS	PASS
7:10:48	135.4	6080.21	224.610942880482	1.05568861603352	38.9699831	127.1338502	PASS	PASS	PASS	PASS
7:10:49	125.5	6075.67	227.473434682999	0.535536848885961	38.9685459	127.1353146	PASS	PASS	PASS	PASS
7:10:50	135	6060.05	226.047940756981	0.953675907320739	38.9671834	127.1368257	PASS	FAIL	PASS	FAIL
7:10:51	131.5	6039.55	228.748183531923	1.66495477317055	38.9657857	127.1383329	PASS	PASS	PASS	PASS
7:10:52	135.3	6018.51	228.850355125141	1.80164773555783	38.9643971	127.1398493	PASS	FAIL	PASS	FAIL
7:10:53	135.1	6003.66	230.233541658393	1.73126049650742	38.9629738	127.1413562	PASS	FAIL	PASS	FAIL
7:10:54	134.4	5996.71	232.728102238827	1.60412367011606	38.9615177	127.1428681	PASS	FAIL	PASS	FAIL
7:10:55	144.1	5996.96	232.238372464448	1.55235919712957	38.9600686	127.1443813	PASS	FAIL	PASS	FAIL
7:10:56	143.5	6003.14	233.335191665016	1.06292899422452	38.9585666	127.1458545	PASS	FAIL	PASS	FAIL
7:10:57	141.4	6004.8	232.806742570591	0.715880496674066	38.9569795	127.1472281	PASS	PASS	PASS	PASS
7:10:58	137.1	5987.86	234.220293522712	0.783312022217982	38.9553391	127.1485492	PASS	FAIL	PASS	FAIL
7:10:59	131.7	5954.23	232.808979270991	0.952208854296933	38.9537234	127.1498548	PASS	FAIL	PASS	FAIL
7:11:0	134.6	5905.54	238.064234833169	1.38913417454002	38.9521355	127.151231	PASS	PASS	PASS	PASS
7:11:1	136.4	5848.59	240.261055953452	1.79639735722326	38.950605	127.1526767	PASS	FAIL	PASS	FAIL
7:11:2	135.3	5792.64	241.918945906282	2.23581703958364	38.9491153	127.154189	PASS	FAIL	PASS	FAIL
7:11:3	129.4	5743.42	244.394422255908		38.9476314	127.1557586	PASS	FAIL	PASS	FAIL

Εικόνα 33: Αποτελέσματα αξιολόγησης από το Smart Falcon (Περίπτωση Β)

Όπως βλέπουμε στις παραπάνω εικόνες το πρόγραμμα δείχνει ότι ο μαθητής δεν περάτωσε την εκπαίδευση αφού σε πολλές φάσεις της πτήσης δεν κατάφερε να κινείται εντός των επιθυμητών παραμέτρων (στήλη Final του πίνακα).

6

Viper Training Station

6.1 Εισαγωγή

Στο δεύτερο κεφάλαιο και στην έρευνα η οποία διεξήχθη ανάμεσα σε ιπτάμενους τόσο της Πολιτικής Αεροπορίας όσο και της Πολεμικής Αεροπορίας ποσοστό 86,6% των ερωτηθέντων απάντησε ότι η διασύνδεση πολλών flight simulator θα ενίσχυε την παρεχόμενη εκπαίδευση σε σημαντικό βαθμό. Σε αυτό το κεφάλαιο θα εξετάσουμε την δημιουργία ενός flight simulator station με συνολικά 8 flight simulators τα οποία συνεργάζονται μεταξύ τους καθώς και 2 θέσεις flight simulator για ελεγκτές ραντάρ. Το flight simulator station θα έχει συνολικά 10 flight simulators τα οποία θα συνεργάζονται μεταξύ τους παρέχοντας την δυνατότητα για τις ρεαλιστικότερες συνθήκες εκπαίδευσης τόσο στους ιπταμένους που το χρησιμοποιούν όσο και στους ελεγκτές ραντάρ που θα συμμετέχουν στις εκάστοτε αποστολές.

6.2 Κατηγοριοποίηση flight simulator σύμφωνα με τον ICAO.

Στο έγγραφο του ICAO Doc. 9625 (ICAO, 2020) με τίτλο “Εγχειρίδιο κριτηρίων για την πιστοποίηση των εκπαιδευτικών συσκευών προσομοίωσης πτήσης” γίνεται κατηγοριοποίηση των flight simulator σε επτά κατηγορίες. Οι κατηγορίες αυτές φαίνονται στον πίνακα που ακολουθεί (Πίνακας 2):

Τύποι FSTD	International Civil Aviation Organization
I	Στον τύπο I το FSTD πρέπει να μπορεί να κάνει πιστή αναπαράσταση του cockpit του αεροσκάφους που αφορά η προσομοίωση. Τα όργανα του cockpit πρέπει να φωτίζονται επαρκώς για την λειτουργία του FSTD και πρέπει να υπάρχουν και ηχητικά μηνύματα τα οποία αφορούν την λειτουργία του αεροσκάφους. Δεν απαιτείται οι συσκευές FSTD αυτής της κατηγορίας να κάνουν προσομοίωση του περιβάλλοντος (περιβάλλον αεροδρομίου, καιρός, μηνύματα ATC κ.α.).

II	Στον τύπο II το FSTD πρέπει να ικανοποιεί όλες τις απαιτήσεις του τύπου I και επιπρόσθετα πρέπει να κάνει προσομοίωση του ATC περιβάλλοντος με χρήση μηνυμάτων και οπτικοακουστικών μέσων.
III	Στον τύπο III το FSTD πρέπει να ικανοποιεί όλες τις απαιτήσεις του τύπου II και επιπρόσθετα πρέπει να κάνει προσομοίωση της κατάστασης του διαδρόμου δηλαδή πρέπει να μπορεί να τον παρουσιάζει ως στεγνό ή βρεγμένο. Δεν είναι απαραίτητο αυτός ο τύπος FSTD να κάνει προσομοίωση του ATC περιβάλλοντος πράγμα που συναντήσαμε στον τύπο II.
IV	Στον τύπο IV το FSTD πρέπει να ικανοποιεί όλες τις απαιτήσεις που ικανοποιούν οι προηγούμενοι τύποι. Πρέπει δηλαδή να μπορεί να κάνει προσομοίωση του περιβάλλοντος διαδρόμου μαζί με την προσομοίωση του ATC περιβάλλοντος (χρήση μηνυμάτων και οπτικοακουστικών μέσων).
V	Στον τύπο V το FSTD πρέπει να ικανοποιεί όλες τις απαιτήσεις του τύπου IV και επιπρόσθετα πρέπει να κάνει προσομοίωση στις συνθήκες διαδρόμου παρουσιάζοντάς τον ως στεγνό, βρεγμένο, παγωμένο και με λιμνάζοντα νερά στην επιφάνειά του. Στον τύπο αυτό εισάγονται και οι δυνατότητες προσομοίωσης των συσκευών προειδοποίησης του αεροσκάφους, η δυνατότητα αίσθησης της κίνησης με οπτικοακουστικά μέσα και η δυνατότητα προσομοίωσης όλων των ειδών βλαβών που αφορούν βλάβες φρένων του αεροσκάφους.
VI	Στον τύπο VI το FSTD πρέπει να ικανοποιεί όλες τις απαιτήσεις του τύπου V και επιπρόσθετα πρέπει να κάνει προσομοίωση στις συνθήκες κίνησης του αεροσκάφους με μηχανικό τρόπο. Επιπρόσθετα με αυτό σε αυτό τον τύπο το FSTD πρέπει να κάνει μία εκτεταμένη προσομοίωση του περιβάλλοντος πτήσης πέρα από την περιοχή του διαδρόμου προσγείωσης και να έχει την δυνατότητα προσομοίωσης αναταράξεων κατά την διάρκεια της πτήσης.
VII	Στον τύπο αυτό το FSTD πρέπει να είναι ικανό για τον τέλειο βαθμό προσομοίωσης πτήσης να μπορεί να προσομοιώνει όλες τις συνθήκες πτήσης, δίνοντας έμφαση στον ρεαλισμό της προσομοίωσης και να διασυνδέεται με άλλα FSTD. Πρόκειται για τον τελειότερο τύπο FSTD.

Πίνακας 2: Κατηγοριοποίηση flight simulator(ICAO, 2020)

6.3 Κατηγοριοποίηση του *Viper Training Center*

Το flight simulator το οποίο προτείνουμε σε αυτή την διπλωματική εργασία μπορούμε εύκολα να το κατηγοριοποιήσουμε ως ένα τύπου V+ FSTD μιας και ικανοποιεί όλες τις απαιτήσεις του τύπου V. Δεν περιορίζεται όμως μόνο σε αυτές μια και με την χρήση του λογισμικού Falcon BMS 4.32 μπορεί να κάνει προσομοίωση του περιβάλλοντος πτήσης πέρα από την περιοχή του διαδρόμου προσγείωσης κάτι το οποίο συναντάμε σε FSTD τύπου VI και φυσικά έχει την δυνατότητα διασύνδεσης με άλλα flight simulator, χαρακτηριστικό του τύπου VII FSTD.

6.4 Περιγραφή *Viper Training Center*

6.4.1 Cockpit

Όπως είδαμε και στις προηγούμενες ενότητες αυτού του κεφαλαίου το cockpit είναι από τα σημαντικότερα στοιχεία τα οποία θα να αναπαριστώνται πιστά από ένα flight simulator. Συγκεκριμένα για το F-16 το οποίο είναι και το αεροσκάφος που θέλει να προσομοιάσει αυτό το flight simulator υπάρχουν πάρα πολλές εταιρείες οι οποίες με σχετικά χαμηλό κόστος προσφέρουν την δυνατότητα πιστών αντιγράφων και πλήρως συμβατών με το λογισμικό Falcon BMS 4.32 (Εικόνα 35).



Εικόνα34: Cockpit Viper Training Center(ViperWing, 2020)

6.4.2 Οθόνη

Το δεύτερο σημαντικότερο στοιχείο που θα πρέπει να ενσωματώνει το flight simulator είναι αυτό της απεικόνισης τόσο του περιβάλλοντος του αεροδρομίου όσο και του περιβάλλοντα χώρου που θα κινείται το αεροσκάφος. Το καλύτερο αποτέλεσμα θα μπορούσαμε να το έχουμε με μία οθόνη σε μορφή θόλου όμως το κόστος για μια τέτοια λύση είναι απαγορευτικό. Η καλύτερη εναλλακτική την οποία θα υιοθετήσουμε και στο flight simulator το οποίο προτείνουμε σε αυτή την διπλωματική είναι αυτό μιας κυρτής οθόνης 70 ιντσών. Το τελικό αποτέλεσμα του cockpit μαζί με την οθόνη φαίνεται στην Εικόνα 36.



Εικόνα35: Οθόνη Viper Training Center(ViperWing, 2020)

6.4.3 Διασύνδεση

Η διασύνδεση των flight simulator μπορεί να γίνει είτε σε τοπικό επίπεδο μέσω τοπικού δικτύου, είτε μέσω Web. Με την χρήση του λογισμικού Falcon BMS 4.32 δεν χρειάζεται να αποκλείσουμε κάποια από τις δύο επιλογές αφού το ίδιο το λογισμικό μας επιτρέπει να χρησιμοποιήσουμε και τις δύο. Αυτό μας δίνει την δυνατότητα να έχουμε τα flight simulator σε οποιονδήποτε χώρο και αυτά να συνεχίσουν να λειτουργούν είτε διασυνδεδεμένα (σε τοπικό δίκτυο ή μέσω web) είτε αυτόνομα, ανάλογα με τις απαιτήσεις του χρήστη.

6.4.4 *Tactical Controller Computer*

Για να γίνει ακόμη περισσότερο ρεαλιστική η εκπαίδευση πρέπει να προσθέσουμε άλλες 2 κονσόλες στο flight simulator center για τακτικούς ελεγκτές. Με αυτό τον τρόπο οι οδηγίες στα αεροσκάφη θα δίνονται από πραγματικούς ελεγκτές πτήσης πράγμα το οποίο θα συμβάλλει στην ρεαλιστικότερη εκπαίδευση των ιπταμένων. Να προσθέσουμε σε αυτό το σημείο ότι το λογισμικό Falcon BMS δίνει την δυνατότητα συμμετοχής ελεγκτών στις πτήσεις και το μόνο που απαιτείται είναι ένας υπολογιστής, έτσι ώστε να μπορεί να συνδεθεί ο ελεγκτής στο εικονικό πεδίο της μάχης.

6.4.5 *Debriefing Station Computer*

Μία από τις σημαντικότερες φάσεις κάθε αποστολής που μπορούμε να εκτελέσουμε στο flight simulator station είναι αυτό της απενημέρωσης της εκάστοτε αποστολής. Για τον λόγο αυτό κρίνεται αναγκαία η δημιουργία ενός debriefing station computer από το οποίο θα μπορεί ο κάθε ιπτάμενος να εκτελέσει απενημέρωση της αποστολής που πέταξε στο flight simulator αναγνωρίζοντας τα δυνατά του σημεία και σημεία που χρήζουν βελτίωσης. Για την λειτουργία του debriefing station computer χρησιμοποιήσουμε το λογισμικό Tacview το οποίο γνωρίζουμε ότι συνεργάζεται με το Falcon BMS 4.32 και μάλιστα δεν έχει κανέναν περιορισμό στο πλήθος των ιπταμένων που μπορούν να εκτελέσουν debriefing ταυτόχρονα. Ο υπολογιστής του debriefing station πρέπει να έχει πρόσβαση στον κεντρικό server που θα τρέχει το Smart Falcon, το οποίο κάνει και την αξιολόγηση του ιπταμένου κατά την διάρκεια της εκπαίδευσης. Έτσι μετά το πέρας κάθε αποστολής ο ιπτάμενος θα έχει σε μόλις λίγα λεπτά τα αποτελέσματα της πτήσης του. Τέλος ο υπολογιστής αυτός θα χρησιμοποιείται από τον εκπαιδευτή για παρακολούθηση της εκπαίδευσης σε πραγματικό χρόνο.

6.4.6 *Flight Simulator Server*

Η δικτυακή φύση του SmartFalcon επιτρέπει την εφαρμογή του στο ήδη υπάρχον δίκτυο της Πολεμικής Αεροπορίας. Για το λόγο αυτό δεν απαιτείται να δημιουργηθεί ξεχωριστός server για το SmartFalcon και το σύστημα των simulators παρά μόνο να του ανατεθούν οι αναγκαίοι πόροι του δικτύου έτσι ώστε να ικανοποιεί όλες τις μη λειτουργικές και λειτουργικές απαιτήσεις. Πρόκειται για ένα server, ο οποίος είναι υπεύθυνος για την διασύνδεση τόσο των flight simulators όσο και των debriefing station computers. Καταλαβαίνουμε ότι αυτός ο server είναι και η “καρδιά” του flight simulator station.

6.4.7 *Πλεονεκτήματα και μειονεκτήματα*

Η εκπαίδευση των ιπταμένων στο Viper Training Station και η αξιολόγησή από το SmartFalcon παρουσιάζει πολλαπλά οφέλη. Αρχικά δίνεται η δυνατότητα στον εκπαιδευόμενο να εκπαιδευτεί σε ένα ρεαλιστικό περιβάλλον με πραγματικούς αντίπαλους και σχηματιζόμενους (άλλοι εκπαιδευόμενοι ή εκπαιδευτές). Η δυνατότητα αυτή υπάρχει και στα υπάρχοντα συστήματα που χρησιμοποιεί η Πολεμική Αεροπορία, με την διαφορά ότι η υλοποίησή της προϋποθέτει την αγορά ισάριθμων συστημάτων εξομοίωσης με τεράστιο κόστος. Λόγω του αυξημένου κόστους, οι Πολεμικές Αεροπορίες συνήθως προμηθεύονται ένα ή δύο εξομοιωτές με αποτέλεσμα, η όποια τακτική εκπαίδευση των ιπταμένων στον εξομοιωτή να περιορίζεται από την χρήση bots που εκτελούν μόνο βασικές ενέργειες. Επιπλέον το γραφικό περιβάλλον των open source εξομοιωτών είναι ίδιο με των πραγματικών εξομοιωτών και αναβαθμίζεται με ευκολία και συνεχώς, κάτι το οποίο δεν συμβαίνει στην περίπτωση των τελευταίων.

Στον αντίποδα, παρόλο που το προτεινόμενο σύστημα εξομοίωσης παρέχει την δυνατότητα ρεαλιστικής εκπαίδευσης σε τακτικό περιβάλλον, δεν έχει την δυνατότητα εξομοίωσης διαδικασιών ανάγκης. Ο λόγος είναι ότι ο προγραμματισμός όλων των συστημάτων ενός αεροσκάφους και των πιθανών βλαβών που δύναται να προκληθούν, ξεπερνά την φύση ενός gaming flight simulator. Επιπρόσθετα, λόγω του απορρήτου των οπλικών συστημάτων ενός μαχητικού, η εκπαίδευση στην χρήση τους μέσω ενός gaming flight simulator είναι σαφώς περιορισμένη.

7

Οικονομική Μελέτη

7.1 Εισαγωγή

Στο δεύτερο κεφάλαιο και στην έρευνα η οποία διεξήχθη ανάμεσα σε ιπτάμενους τόσο της Πολιτικής Αεροπορίας όσο και της Πολεμικής Αεροπορίας ποσοστό 86,6% των ερωτηθέντων απάντησε ότι η διασύνδεση πολλών flight simulator θα ενίσχυε την παρεχόμενη εκπαίδευση σε σημαντικό βαθμό. Ως εκ τούτου κρίνεται μη ρεαλιστική οικονομική μελέτη για ένα και μόνο flight simulator το οποίο δεν είναι διασυνδεδεμένο και δεν συνεργάζεται με άλλα flight simulator. Σε αυτό το κεφάλαιο θα εξετάσουμε αρχικά το κόστος ανά ώρα πτήσης στα υπάρχοντα flight simulator πολλά εκ των οποίων δεν είναι διασυνδεδεμένα και στη συνέχεια θα το συγκρίνουμε με το κόστος πτήσης στο flight simulator station το οποίο προτείνουμε σε αυτή τη διπλωματική.

7.2 Κόστος χρήσης υπαρχόντων flight simulator

Σύμφωνα με την εταιρεία FAAC Military, μια από τις μεγαλύτερες εταιρίες που προσφέρει εκπαίδευση τόσο στον στρατό των Η.Π.Α. όσο και σε άλλους στρατούς ανά τον κόσμο, το κόστος πτήσης με ένα πραγματικό F-16 είναι \$7500/ώρα. Σύμφωνα πάντα με την ίδια εταιρία, το κόστος πτήσης στα flight simulator training center τα οποία διαθέτει η ίδια είναι \$900/ώρα για το κάθε ένα flight simulator που διαθέτει. Τα flight simulator της εταιρίας είναι κατηγορίας V, είναι δηλαδή παρόμοια με το flight simulator το οποίο προτείνουμε σε αυτή την διπλωματική. Αυτό μας δίνει μία γενική εικόνα για το κόστος εκπαίδευσης με military type flight simulator. Ο λόγος που δεν θα χρησιμοποιήσουμε το πραγματικό κόστος εκπαίδευσης είναι γιατί αποτελεί διαβαθμισμένη πληροφορία. (FAAC Military, 2020)

7.3 Κόστος εγκατάστασης Viper Training Center

Όπως είδαμε στο προηγούμενο κεφάλαιο δεν αρκεί ένα μόνο του flight simulator για την εκπαίδευση των ιπταμένων. Το flight simulator center πρέπει να διαθέτει 8 διασυνδεδεμένα flight simulators, 2 tactical controller computers, 10 debriefing station

computers, και 1 flight simulator server. Τα εκτιμώμενα κόστη για τα παραπάνω υλικά φαίνονται στον πίνακα που ακολουθεί (Πίνακας 3). Για τον υπολογισμό τους χρησιμοποιήθηκε η ισοτιμία 1 Ευρώ = 1,1 Δολάρια Αμερικής και φόρος προστιθέμενης αξίας (ΦΠΑ) 24%.

Είδος	Ποσότητα	Τιμή Μονάδας (Ευρώ)	Σύνολο (Ευρώ)
F-16 Inspired Simulator Frame	8	€ 3380,69 [5]	€ 27.045,52
F-16 Inspired Ejection Seat	8	€ 1.689,78 [5]	€ 13.518,25
Curved Screen	8	€ 2.624,32 [5]	€ 20.994,56□
Tactical Controller Computer	2	€ 2.900	€ 5.800
DebriefingStation Computer	8	€ 2.000	€ 16.000
Flight Simulator Server	1	€ 10.000	€ 10.000
Σύνολο			€ 93.358,33

Πίνακας 3: Εκτιμώμενο κόστος ViperTrainingCenter

Από τον παραπάνω πίνακα φαίνεται ότι το κόστος της επένδυσης για ένα Flight Simulator Center διαμορφώνεται σε **€ 93.358,33**. Δεν βλέπουμε πουθενά κόστος για λογισμικό γιατί το λογισμικό που θα χρησιμοποιηθεί είναι Open Source. Επιπρόσθετα σε όλο το hardware που επιλέχθηκε, προστέθηκε στην τιμή και εγγύηση 5 ετών. Αν τέλος προσθέσουμε και συντελεστή απόκλισης της τάξης του 30% το τελικό κόστος του Flight Simulator Center διαμορφώνεται στα **€ 121.365,829**□

7.4 Κόστος χρήσης Viper Training Center

Για να δούμε το κόστος χρήσης του Flight Simulator Center ανά ώρα πτήσης πρέπει να προσδιορίσουμε αρχικά τον χρονικό ορίζοντα χρήσης του. Όπως προαναφέραμε όλο το hardware έχει υπολογιστεί ότι θα αγοραστεί με 5ετή εγγύηση λειτουργίας. Για αυτό τον λόγο οι υπολογισμοί μας θα βασιστούν σε αυτό το χρονικό διάστημα. Οι πτήσεις που μπορούν να εκτελεστούν σε αυτό το χρονικό διάστημα θα υπολογιστούν με βάση 20 εργάσιμων ημερών και λειτουργία του για 8 ώρες. Δηλαδή το

Flight Simulator Center για 5 χρόνια θα αποδώσει 76.800 ώρες πτήσης. Διαιρώντας το κόστος κατασκευής του με τις ώρες πτήσης βλέπουμε ότι κάθε ώρα κοστίζει μόνο € 1,58. Στο κόστος αυτό όμως δεν έχει προστεθεί το κόστος της ηλεκτρικής κατανάλωσης του εξοπλισμού καθώς αυτό διαμορφώνεται ανάλογα με τον τρόπο παραγωγής του θα μπορούσαμε όμως με ασφάλεια να πούμε ότι αυτό το κόστος δεν θα ξεπερνά σε καμία περίπτωση τα € 0,5. Συνοψίζοντας το τελικό κόστος ανά ώρα πτήσης διαμορφώνεται στα **€ 2,08**. Κόστος κατά πολύ μικρότερο από το κόστος πτήσης των εξομοιωτών που χρησιμοποιούνται μέχρι σήμερα.

8

Προοπτικές-Συμπεράσματα

8.1 Προοπτικές

Είναι εύκολα κατανοητό, ότι η παραμετροποίηση όλων των μεταβλητών μιας πτήσης μαχητικού αεροσκάφους είναι ένα δύσκολο και χρονοβόρο έργο για την δυναμική αυτής της διπλωματικής εργασίας. Η δυνατότητα επέκτασης του συγκεκριμένου project καθιστά εφικτή την επίτευξη δραματικής μείωσης των απαιτούμενων υλικών και ανθρώπινων πόρων και την ταυτόχρονη αύξηση της ποιότητας της παρεχόμενης εκπαίδευσης. Τέτοια συστήματα είναι εύκολο να εφαρμοστούν τόσο στην πολιτική όσο και στην Πολεμική Αεροπορία. Η μεγάλη πρόοδος που έχει σημειωθεί τα τελευταία χρόνια στον τομέα της τεχνητής νοημοσύνης επιτρέπει τη χρήση αλγορίθμων, ευφυών πρακτόρων, νευρωνικών δικτύων και άλλων τεχνολογιών, μέσω των οποίων όχι μόνο μπορούν να παρακολουθούνται πολλές μεταβλητές αλλά υπάρχει και η δυνατότητα αυτόματης αναπροσαρμογής τους βάσει των απαιτήσεων του καθενός εκπαιδευόμενου ξεχωριστά. Με αυτό τον τρόπο κάθε εκπαιδευόμενος θα λαμβάνει εξατομικευμένη εκπαίδευση, θα αξιολογείται και θα αναπροσαρμόζεται η εκπαίδευσή του βάσει της απόδοσης του αυτόματα, χωρίς την παρέμβαση κάποιου εκπαιδευτή. Τα συστήματα αυτά χαρακτηρίζονται ως ευφυή διδακτικά συστήματα (intelligent tutoring systems) και ήδη χρησιμοποιούνται από τις Ένοπλες Δυνάμεις των Ηνωμένων Πολιτειών για την εκπαίδευση των στελεχών τους. Η εισαγωγή τέτοιων συστημάτων σε ήδη υπάρχοντες εξομοιωτές πτήσεων, αφαιρεί τον παράγοντα “άνθρωπος” από το σύστημα εξομοιωτής- εκπαιδευόμενος με πολλαπλά πλεονεκτήματα. Αφενός επιτυγχάνεται μείωση του κόστους και αφετέρου απαλείφονται οι πεπερασμένες νοητικές δυνατότητες του ανθρώπου και η επίδραση τους στην εκπαίδευση.

8.2 Συμπεράσματα

Από την εξέταση των απαντήσεων του ερωτηματολογίου, διαφάνηκε ξεκάθαρα η ανάγκη για αναβάθμιση της εκπαίδευσης στον εξομοιωτή πτήσεων και η εμπιστοσύνη στην ύπαρξη ενός λογισμικού που θα λειτουργεί επικουρικά ή εντελώς αυτόνομα στην εκπαιδευτική διαδικασία.

Η δημιουργία και εφαρμογή του Smart Falcon ανέδειξε τις δυνατότητες από την χρήση του και τις ανεξάντλητες δυνατότητες αναβάθμισης και εξέλιξής του. Μπόρεσε με επιτυχία να αξιολογήσει την απόδοση μικρού αριθμού ιπταμένων σε συγκεκριμένες παραμέτρους πτήσης και έγινε κοινώς αποδεκτό ως ένα πολύ χρήσιμο συμπληρωματικό εργαλείο της εκπαίδευσης στον εξομοιωτή πτήσεων. Κατά την διάρκεια της ανάπτυξής του, το σημαντικότερο πρόβλημα που παρουσιάστηκε ήταν αυτό της μετατροπής των αρχείων .dvr του FalconBMS, σε αξιοποιήσιμα αρχεία .csv. Το πρόβλημα αυτό υπερκεράστηκε μέσω της χρήσης ενός τριτεύοντος open source προγράμματος, του TacView. Η παρούσα έκδοση του Smart Falcon αποτελεί ένα πρωτότυπο με περιορισμένες δυνατότητες και υλοποιήθηκε καθαρά για λόγους επίδειξης των δυνατοτήτων της πλήρως λειτουργικής έκδοσης. Για το λόγο αυτό δεν αξιολογεί το σύνολο των παραμέτρων μιας πτήσης και δεν έχει παραμετροποιηθεί ένα συγκεκριμένο profile αποστολής (π.χ. τροχοδρόμηση, απογείωση, εναέρια μάχη, προσγείωση) ή κάποια συγκεκριμένη άσκηση αέρος (π.χ. ακροβατικές ασκήσεις). Μπορεί όμως να χρησιμοποιηθεί ως βάση για την εξέλιξή της. Η πλήρως λειτουργική έκδοση θα αξιολογεί όλα τα είδη των ασκήσεων και των αποστολών που μπορούν να εκτελέσουν οι ιπτάμενοι. Η φιλοσοφία του Smart Falcon το καθιστά άμεσα εφαρμόσιμο σε open source εξομοιωτές όπως το FalconBMS, αλλά δεν είναι αξιοποιήσιμο από τα εν χρήσει συστήματα εξομοίωσης. Η δημιουργία προγραμμάτων που θα συνεργάζονται με τα τελευταία απαιτεί πρόσβαση στον πηγαίο κώδικα των μαχητικών αεροσκαφών, ο οποίος δεν είναι συνήθως αποδεσμεύσιμος από τις κατασκευάστριες εταιρείες.

Είναι σαφές ότι, είναι πλέον εφικτή η απαλοιφή των μειονεκτημάτων που παρουσιάζει ο άνθρωπος-εκπαιδευτής από την εξίσωση εξομοιωτής-εκπαιδευόμενος και η εξαιρετικά σημαντική μείωση του κόστους της εκπαίδευσης του ιπτάμενου προσωπικού. Λόγω των διευρυμένων δυνατοτήτων αντίστοιχων συστημάτων που υπάρχουν ήδη σε χρήση, είναι πλέον μονόδρομος η στροφή της εκπαίδευσης προς αυτή την κατεύθυνση. Εφαρμόζοντας τεχνολογίες όπως τα νευρωνικά δίκτυα και η εικονική πραγματικότητα, είναι πιθανό να οδηγήσουν σε μια εκπαίδευση εξ ολοκλήρου βασιζόμενη στην εξομοίωση.

Βιβλιογραφία

- Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The unified modeling language user guide*. Boston, MA: Addison Wesley.
- Department Of Defense. (1994, 12). <http://everyspec.com>. Ανάκτηση από <http://everyspec.com>: http://everyspec.com/MIL-STD/MIL-STD-0300-0499/MIL-STD-498_25500/
- FAAC Military. (2020). *FAAC Military-MILITARY SIMULATION TRAINING SOLUTIONS FOR COMBAT AND TACTICS*. Ανάκτηση από <https://www.faac.com/military/>
- Facebook Inc. (2020, 06 20). <https://reactjs.org/>. Ανάκτηση από <https://reactjs.org/>: <https://reactjs.org/>
- ICAO, I. (2020, 06 20). *Doc 9625, Manual of Criteria for the Qualification of Flight Simulation Training Devices, Volume 1 — Aeroplanes*. Ανάκτηση από <http://www.icscc.org.cn/>: <http://www.icscc.org.cn/upload/file/20190102/Doc.9625-EN%20Manual%20of%20Criteria%20for%20the%20Qualification%20of%20Flight%20Simulation%20Training%20Devices%20Volume%20I%20-%20Aeroplanes.pdf>
- Jones, R. M., Laird, J. E., Nielsen, P. E., Coulter, K. J., Kenny, P., & Koss, F. V. (1999, Spring). Automated Intelligent Pilots for Combat Flight Simulation. *AI Magazine*, σσ. 1-4.
- Remolina, E., Ramachandran, S., Fu, D., Stottler, R., & Howse, W. R. (2004). *Intelligent Simulation-Based Tutor for Flight Training*. Ανάκτηση από <https://discover.dtic.mil/>: <https://apps.dtic.mil/sti/citations/ADA439800>
- Russ Ferguson, C. H. (2013). *Beginning JavaScript with DOM Scripting and Ajax*. Berkeley, CA: Apress.
- ViperWing. (2020). <https://viperwing.com/>. Ανάκτηση από <https://viperwing.com/>.
- Βεσκούκης, Β. (2015). *Μοντέλα κύκλου ζωής λογισμικού*. Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών.