

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Athens MBA**

**Ειρήνη Καραγιάννη**

**Εφαρμογή της Lean Φιλοσοφίας σε εταιρικό Τμήμα Ανάπτυξης Λογισμικού**

**Έτος Εκπόνησης 2019-2020**

**Επιβλέπων Καθηγητής Βρασίδης Λεώπουλος**

## ΔΗΛΩΣΗ ΕΚΠΟΝΗΣΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

«Δηλώνω υπεύθυνα ότι η συγκεκριμένη μεταπτυχιακή εργασία για τη λήψη του Μεταπτυχιακού Διπλώματος Ειδίκευσης στη Διοίκηση Επιχειρήσεων, έχει συγγραφεί από εμένα προσωπικά και δεν έχει υποβληθεί ούτε έχει εγκριθεί στο πλαίσιο κάποιου άλλου μεταπτυχιακού ή προπτυχιακού τίτλου σπουδών, στην Ελλάδα ή στο εξωτερικό.

Η εργασία αυτή έχοντας εκπονηθεί από εμένα, αντιπροσωπεύει τις προσωπικές μου απόψεις επί του θέματος. Οι πηγές στις οποίες ανέτρεξα για την εκπόνηση της συγκεκριμένης μεταπτυχιακής αναφέρονται στο σύνολό τους, δίνοντας πλήρεις αναφορές στους συγγραφείς, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο».

Όνοματεπώνυμο  
Ειρήνη Καραγιάννη

Υπογραφή  


## **ΕΥΧΑΡΙΣΤΙΕΣ**

Ευχαριστώ την οικογένειά μου και τους συναδέλφους που με στήριξαν αυτά τα δημιουργικά μεν δύσκολα δε 2,5 χρόνια.

Ευχαριστώ ιδιαίτερω τον κ. Τσοσκούνογλου για την υπομονή και την καθοδήγηση.

## ΠΕΡΙΕΧΟΜΕΝΑ

ΔΗΛΩΣΗ ΕΚΠΟΝΗΣΗΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ.....	2
ΕΥΧΑΡΙΣΤΙΕΣ.....	3
ΠΕΡΙΕΧΟΜΕΝΑ .....	4
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ .....	9
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ .....	10
ΣΥΝΟΨΗ .....	11
ABSTRACT.....	15
ΚΕΦΑΛΑΙΟ 1. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	19
1.1 Interchangeable parts .....	19
1.2 Interchangeable (Αναλώσιμοι) Εργάτες. ....	21
1.3 Μαζική παραγωγή αυτοκινήτων.....	22
1.4 Το Σύστημα Παραγωγής Toyota (Toyota Production System). ....	23
1.4.1 Τα πρόσωπα πίσω από το TPS (Toyota Production System). ....	24
1.5 Ιαπωνική εξάπλωση.....	25
1.6 Lean και υπηρεσίες. ....	26
ΚΕΦΑΛΑΙΟ 2 . ΤΙ ΕΙΝΑΙ LEAN.....	27
2.1 Η έννοια του Lean. ....	27
2.2 Just-In-Time / Autonomation(Jidoka). ....	29
2.3 Βασικές αρχές Lean σκέψης. ....	31
2.4 Βασικά Lean «εργαλεία».....	33
2.5. Κριτικές για το Lean.....	39

2.5.1	Απώλεια θέσεων.....	39
2.5.2	Δυσκολία εφαρμογής.....	40
ΚΕΦΑΛΑΙΟ 3. LEAN SERVICES / LEAN IT .....		41
3.1	Services και Lean.....	41
3.2	Παρούσα κατάσταση στο IT.....	43
3.3	Οπτική Business .....	44
3.4	Οπτική IT.....	46
3.5	Πώς το Lean IT δημιουργεί αξία.....	49
3.6	Lean και Green IT.....	50
ΚΕΦΑΛΑΙΟ 4 . LEAN ΑΡΧΕΣ ΣΤΟ SW DEVELOPMENT .....		51
4.1	Το SW Development ως προϊόν.....	51
4.2	Εξάλειψη της σπατάλης.....	52
4.3	Εστίαση στην ποιότητα.....	55
4.4	Δημιουργία Γνώσης.....	56
4.5	Αναβολή Δέσμευσης (Defer Commitment).....	57
4.6	Ταχεία Παράδοση (Deliver Fast).....	58
4.7	Σεβασμός στους Ανθρώπους.....	59
4.8	Βελτιστοποίηση του Συνόλου (Optimize the Whole).....	61
ΚΕΦΑΛΑΙΟ 5. ΔΗΜΙΟΥΡΓΙΑ ΑΞΙΑΣ ΓΙΑ ΤΟΝ ΠΕΛΑΤΗ .....		62
5.1	Lean Consumption.....	62
5.2	Η έννοια «Πελάτης» στο εσωτερικό SW Development.....	65
5.3	Συνεργασία Business – IT.....	66
5.4	Συμπεράσματα.....	67
ΚΕΦΑΛΑΙΟ 6. ΣΠΑΤΑΛΕΣ ΣΤΟ SW DEVELOPMENT .....		68

6.1	Εισαγωγή. ....	68
6.2	Ημιτελής δουλειά. (Partially Done Work).....	69
6.3	Υπερβάλλουσα λειτουργικότητα. (Extra Features).....	70
6.4	Επανέυρεση λύσης (Relearning).....	71
6.5	Μεταφορά εργασίας (Handoffs). ....	72
6.6	Εναλλαγή εργασίας (Task switching ). ....	72
6.7	Καθυστερήσεις-Αναμονές(delays).....	73
6.8	Ελαττωματικός κώδικας (Defects). ....	74
6.9	Υποαπασχόληση. ....	74
6.10	Αποτύπωση του Value Stream. ....	75
6.11	Διατήρηση της απλότητας.....	76
<b>ΚΕΦΑΛΑΙΟ 7. ΟΙ ΑΝΘΡΩΠΟΙ ΣΤΗ LEAN ΦΙΛΟΣΟΦΙΑ.....</b>		<b>78</b>
7.1	Εισαγωγή (Lean system of management). ....	78
7.2	Σύστημα Deming. ....	78
7.3	Λόγοι αποτυχίας των προγραμμάτων μετασχηματισμού. ....	81
7.4	Η έννοια της ομάδας. ....	82
7.5	Εξειδίκευση. ....	83
7.6	Ηγεσία.....	84
7.7	Αξιολόγηση. ....	85
<b>ΚΕΦΑΛΑΙΟ 8. ΟΔΗΓΟΣ LEAN ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΣΤΟ IT ΚΑΙ ΣΤΟ SW.....</b>		<b>86</b>
8.1	Πυλώνες Lean μετασχηματισμού. ....	86
8.2	Εκπαίδευση υπευθύνων. ....	86
8.3	Ο τρόπος σκέψης. ....	87
8.4	Μετρήσεις. ....	87
8.5	Βασικά σημεία Lean μετασχηματισμού στο SW Development.....	90

8.6	Επίλογος.....	93
ΚΕΦΑΛΑΙΟ 9. CASE STUDY.....		94
9.1	Εισαγωγή.....	94
ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ .....		95
9.2	Οργάνωση / Λειτουργία.....	95
9.3	Χρήστες.....	97
9.4	Υπάρχουσες διαδικασίες.....	98
ΠΡΟΤΑΣΕΙΣ ΕΦΑΡΜΟΓΗΣ LEAN .....		99
9.5	Kaizen Initialization Project .....	99
9.6	Καθημερινό Kaizen.....	100
9.7	Ορισμός υπευθύνου για το καθημερινό Kaizen. ....	100
9.8	Εντοπισμός Βασικών Προβλημάτων - Εντοπισμός Σπαταλών - Προτεινόμενες Λύσεις .....	101
9.8.1	Απροσδόκητος τερματισμός λειτουργίας .....	102
9.8.2	Διαφορετική εικόνα στα συστήματα Παραγωγικό/Development/QA .....	104
9.8.3	Εξουθένωση συναδέλφων/ κακό εργασιακό κλίμα .....	106
9.8.4	Παρενέργειες υλοποιημένων έργων.....	109
9.8.5	Δημιουργία πολλών custom προγραμμάτων.....	111
9.8.6	Ανακεφαλαίωση.....	113
9.9	Ευκολία Υλοποίησης και Χρονοδιάγραμμα Προτεινόμενων Λύσεων.....	114
9.10	KPIs για την Υποστήριξη της εφαρμογής του Lean .....	115
9.11	Τελικά συμπεράσματα .....	116
ΠΑΡΑΡΤΗΜΑ I.....		118
Manifesto for Agile Software Development .....		118
ΠΑΡΑΡΤΗΜΑ II.....		120

Σπατάλες στο IT και στο SW Development..... 120

ΒΙΒΛΙΟΓΡΑΦΙΑ ..... 125



## **ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ**

Σχήμα 1. Η θάλασσα των αποθεμάτων .....	28
Σχήμα 2. Automation (Jidoka) .....	30
Σχήμα 3. Οι 5 Lean αρχές .....	31
Σχήμα 4. Παράδειγμα Lean flow .....	32
Σχήμα 5. Why-Why analysis.....	35
Σχήμα 6. Fishbone diagram.....	36
Σχήμα 7. Kaizen.....	37
Σχήμα 8. Οι 5 βασικές αρχές του Kaizen .....	37
Σχήμα 9. Waterfall approach .....	53
Σχήμα 10. Κόστος πολυπλοκότητας.....	70
Σχήμα 11. Value stream map .....	77
Σχήμα 12. Οργανόγραμμα Τμήματος .....	96

## **ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ**

Πίνακας 1. 5 S.....	34
Πίνακας 2. Μεθοδολογίες Επίλυσης Προβλημάτων .....	36
Πίνακας 3. Η κουλτούρα του Kaizen .....	38
Πίνακας 4. CHAOS Report .....	44
Πίνακας 5. Διαφορές Lean Manufacturing / Lean Development.....	52
Πίνακας 6. Σπατάλες στο SW Development .....	68
Πίνακας 7. Συνοπτικός Πίνακας Προβλημάτων .....	113

## ΣΥΝΟΨΗ

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι η εμβάθυνση στην έννοια της Lean (Λιτής) φιλοσοφίας στην Πληροφορική (IT) και στην ανάπτυξη λογισμικού ( SW Development) . Ο βασικότερος στόχος είναι να δοθεί το πλαίσιο μέσω του οποίου μπορεί να γίνει εφαρμογή του Lean σε ένα τμήμα Πληροφορικής που ασχολείται με την ανάπτυξη λογισμικού ( ανάπτυξη κώδικα / παραμετροποίηση ERP ).

Το υλικό συγκεντρώθηκε από βιβλιογραφικές και διαδικτυακές αναφορές και εμπλουτίστηκε με παραδείγματα από προσωπικές εργασιακές εμπειρίες.

Το Lean ως φιλοσοφία και εφαρμογή δημιουργήθηκε από την εταιρεία Toyota μετά το 2<sup>ο</sup> Παγκόσμιο Πόλεμο. Το σύστημα του Lean Production (Λιτής Παραγωγής) που εφάρμοσε η Toyota ώστε να αντιμετωπίσει τα οικονομικά προβλήματα μετά τον πόλεμο, ονομάστηκε TPS (Toyota Production System).

Η Toyota δεν είχε τη δυνατότητα για μαζική παραγωγή και αξιοποίηση οικονομιών κλίμακας. Η αγορά στην Ιαπωνία ήταν κατεστραμμένη και μικρή. Έτσι προσπάθησε να δημιουργήσει κέρδος εξαλείφοντας οποιαδήποτε σπατάλη (π.χ. αποθέματα) σε όλο το μήκος της αλυσίδας αξίας (από τη στιγμή της παραγγελίας από τον πελάτη έως την πληρωμή της).

Η επαναστατική φιλοσοφία της Toyota βρήκε εφαρμογή σε πολλές παραγωγικές βιομηχανίες. Επεκτάθηκε όμως και στο χώρο των υπηρεσιών όπως Υγεία, Αεροπορικές εταιρείες, Logistics, Πληροφορική κλπ.

Παρόλο που οι Lean αρχές είναι ίδιες, υπάρχουν μερικές βασικές διαφορές ανάμεσα στην παραγωγή και τις υπηρεσίες και πιο συγκεκριμένα:

- πρώτα απ' όλα το υλικό στις υπηρεσίες δεν είναι χειροπιαστό αλλά άυλο.
- στην παραγωγή οι σπατάλες, οι αναμονές, η συμφόρηση (bottleneck) είναι άμεσα ορατές σε αντίθεση με τις υπηρεσίες. (π.χ. ένας σωρός από αντικείμενα που περιμένουν κάποιον να τα επεξεργαστεί ή εργάτες που δεν κάνουν τίποτα περιμένοντας να ολοκληρωθεί προηγούμενο βήμα)
- ο ανθρώπινος παράγοντας παίζει ακόμα μεγαλύτερο ρόλο στις υπηρεσίες σε σχέση με την παραγωγή, διότι υπάρχει άμεση επαφή με τον πελάτη και η διαδικασία έχει μεγάλο αντίκτυπο στην ικανοποίηση ή μη του πελάτη. Τη διαδικασία παραγωγής δεν την αντιλαμβάνεται άμεσα ο πελάτης, παρά μόνο μέσα από την εμπειρία του

προϊόντος (π.χ. αγορά τηλεφώνου). Στις υπηρεσίες όμως τη «διαδικασία παραγωγής» του άυλου προϊόντος τη «ζει» ο πελάτης, ως ένα βαθμό τουλάχιστον, και συμμετέχει άμεσα σε αυτήν (π.χ. τραπεζικό δάνειο). Θα μπορούσε λοιπόν κάποιος να πει πως στις υπηρεσίες η διαδικασία είναι το προϊόν.

- υπάρχει μεγάλη διαφοροποίηση στο ανθρώπινο δυναμικό σε γνώσεις/εξειδίκευση/ικανότητες και το προϊόν δεν είναι τυποποιημένο, με αποτέλεσμα να δυσχεραίνεται η τυποποίηση των διαδικασιών.

Στις υπηρεσίες όπως η Πληροφορική και η Ανάπτυξη Λογισμικού, συνήθως η συμφόρηση γίνεται αντιληπτή μόνο όταν οι υπάλληλοι παραπονιούνται για το φόρτο εργασίας. Η σπατάλη κρύβεται στις διαδικασίες μέσω των οποίων γίνεται η ροή των δεδομένων (data flow) και παράλληλα η αδράνεια των υπαλλήλων μπορεί εύκολα να κρυφτεί πίσω από ένα PC. Για τους παραπάνω λόγους είναι πολύ σημαντική η λεπτομερειακή αποτύπωση όλων των βημάτων των διαδικασιών, ώστε να αποκαλυφθούν τα σημεία σπατάλης.

Η Λιτή(Lean) Παραγωγή σε σχέση με τη μαζική, χρησιμοποιεί λιγότερους ανθρώπους, χώρους, μηχανήματα, χρόνο (βελτίωση διαδικασιών), αποθέματα. Βελτιώνει επίσης μαζί με την παραγωγικότητα και την ποιότητα, μειώνει τους χρόνους αναμονής οπουδήποτε μπορεί αυτό να γίνει και αποδεσμεύει παραγωγικούς πόρους.

Η βασική φιλοσοφία του Lean, τόσο στην παραγωγή όσο και στις υπηρεσίες, είναι η ίδια δηλ. η μεγιστοποίηση της αξίας που επιθυμεί (και για την οποία πληρώνει) ο πελάτης, με ταυτόχρονη εξάλειψη της σπατάλης παραγωγικών πόρων.

Θα μπορούσαμε να πούμε ότι Lean σημαίνει:

- Την πεποίθηση ότι υπάρχει πάντα ένας καλύτερος τρόπος να γίνει μια συγκεκριμένη εργασία/διαδικασία
- Τη συνεχή προσπάθεια εύρεσης και εξάλειψης της σπατάλης
- Τη μείωση του χρόνου των διαδικασιών
- Τη συμμετοχή όλων των εμπλεκόμενων στη βελτίωση των διαδικασιών και του εργασιακού περιβάλλοντος (περιλαμβάνοντας και τους εργάτες (τους προγραμματιστές στο SW ) που πραγματικά κάνουν τη δουλειά).
- την προληπτική (proactive) λειτουργία μέσω πρόληψης των λαθών(mistake proof)

Όσον αφορά την εφαρμογή του Lean στην Πληροφορική και ειδικότερα στην Ανάπτυξη Λογισμικού (SW Development), σίγουρα υπάρχει πρόσφορο έδαφος. Τα στοιχεία από μελέτες δηλώνουν ότι μεγάλα ποσοστά των projects Πληροφορικής ακυρώνονται,

παρουσιάζουν καθυστερήσεις, μειωμένες λειτουργίες σε σχέση με τις αρχικές προδιαγραφές και λάθη. Όλα αυτά οδηγούν σε αυξημένα κόστη, καθυστερήσεις, άσχημο και στρεσογόνο εργασιακό περιβάλλον και δυσαρεστημένους πελάτες.

Από τη βιβλιογραφία, αλλά και την προσωπική μου εμπειρία, διαπιστώνεται μεγάλη δυσκολία προσαρμογής του κόσμου της Πληροφορικής (IT) στη Lean κουλτούρα, διότι έρχεται σε αντίθεση με την υπάρχουσα τάξη πραγμάτων. Η Lean φιλοσοφία απαιτεί πειθαρχία, εγρήγορση, οργάνωση και να μην αφήνονται τα πράγματα στην τύχη τους.

Στο Case Study του κεφ. 9 γίνεται διερεύνηση της δυνατότητας εφαρμογής του Lean και εξετάζονται τα οφέλη από την εφαρμογή του, σε ένα Τμήμα Ανάπτυξης Λογισμικού SAP. Περιγράφεται η υπάρχουσα κατάσταση και τα προβλήματα, και παρατίθενται συγκεκριμένες προτάσεις για τη βελτίωση των διαδικασιών και την αύξηση της αξίας προς τον πελάτη. Οι προτάσεις αυτές βασίζονται στο Kaizen (τη συνεχή βελτίωση) και στη μείωση της σπατάλης μέσα στο value stream (αλυσίδα αξίας), διότι θεωρώ πως τα δύο αυτά σημεία είναι η καρδιά του Lean.

Η υπάρχουσα κατάσταση στο συγκεκριμένο τμήμα, όπως στις περισσότερες εταιρείες, είναι αρκετά δύσκολη. Υπάρχουν πολλά αιτήματα αλλαγών/ενισχύσεων/projects τα οποία προκαλούν αναστάτωση και άγχος στους υπαλλήλους του τμήματος οι οποίοι προσπαθούν να ανταπεξέλθουν σε όλα όσο γίνεται πιο γρήγορα. Στην προσπάθεια αυτή είναι λογικό να συμβαίνουν λάθη και παραλείψεις. Επίσης λόγω της πίεσης δε δίνεται ο απαραίτητος χρόνος στον αρχικό σχεδιασμό των λύσεων, ούτε στην εύρεση των πραγματικών αιτιών των προβλημάτων, με αποτέλεσμα οι λύσεις να είναι σε μεγάλο βαθμό πυροσβεστικές (reactive και όχι proactive). Στο συγκεκριμένο τμήμα, αλλά και γενικότερα σε οποιοδήποτε SW Τμήμα, οι βασικότερες προτάσεις για τη βελτίωση της υπάρχουσας κατάστασης, είναι :

- η συγγραφή **λιγότερου** και **ποιοτικότερου** κώδικα που θα μεταφέρεται στην παραγωγή **χωρίς καθυστέρηση**.
- να δίνεται περισσότερος χρόνος στο στάδιο του σχεδιασμού ώστε να βρίσκονται οι πραγματικές αιτίες των προβλημάτων και οι λύσεις τους και να μην υπάρχουν επαναλαμβανόμενα προβλήματα. Μια τέτοια λύση θα μπορούσε να είναι η εκπαίδευση των χρηστών ώστε να μην κάνουν λάθη και να μη ζητάνε συνέχεια βοήθεια, ή/και η δημιουργία ελέγχων (validations) ώστε να μην επιτρέπεται η δημιουργία λανθασμένων κινήσεων (mistake proof σύστημα).

- να μην επιτρέπεται να μένει μια εργασία ημιτελής (δηλ. να γίνονται αλλαγές που δεν περνάνε στην παραγωγή).
- η γνώση που δημιουργείται σε προσωπικό επίπεδο να διαχέεται και να γίνεται κτήμα του οργανισμού ( π.χ. με κατάλληλη ενημέρωση του εταιρικού portal )
- να μειωθούν οι διαδικασίες εγκρίσεων αλλαγών που δεν προσφέρουν πραγματική αξία.
- να δημιουργηθούν κατάλληλα KPIs που θα υποστηρίζουν την πραγματική (lean) βελτίωση των διαδικασιών.

Το βασικότερο συμπέρασμα είναι πως το Lean δεν είναι «μασημένη τροφή» που ο καθένας μπορεί να μιμηθεί ή να αντιγράψει. Κάθε οργανισμός πρέπει μόνος του να ανακαλύψει το πώς και πού μπορεί να εφαρμόσει όσα υποστηρίζει και πρεσβεύει το Lean. Ειδικά στο χαοτικό κόσμο ενός Τμήματος Ανάπτυξης Λογισμικού, τα βήματα εφαρμογής του Lean, όπως αυτά που προτείνω, μπορούν να αποδώσουν ορατούς καρπούς.

## **ABSTRACT**

The purpose of the current thesis is delving in the meaning of Lean philosophy in Information Technology and Software Development. The principal aim is the creation of the guidelines needed to implement Lean in an Informatics department that engages in code development and customization of ERP System.

The material was gathered from books and online sources, but also enriched with examples from personal working experience.

Lean, as a philosophy and implementation, was invented from Toyota incorporation after the Second World War. The Lean Production system, which was implemented by Toyota in order to combat its post war financial problems, was named TPS (Toyota Production System).

Toyota did not have the capability of mass production and exploitation of scale economies, as Japan's market was small and damaged. Therefore it attempted to increase income by eliminating any waste (e.g. stock) all through its chain of value (from the moment of the customer's order until its fulfilment).

Toyota's revolutionary philosophy proved to be effective in many production industries. It expanded, however, to the service businesses, like Health, Air Service, Logistics, Informatics, etc.

Although core Lean ideas remain the same, there are some important differences between production and services. More specifically:

- first of all material in service industries is intangible.
- in production, wastes, delays, congestion (bottleneck) are immediately spotted, unlike service industries. (For example a heap of objects that expect processing, or inactive workers, waiting for completion of previous steps)
- human factor has a greater effect in industries of service, in comparison with production, because there is direct contact with the customer and the process significantly affects whether he will be satisfied or not. Whereas, the production procedure is only perceived through the experience of the product (e.g. phone purchase). In service though the customer "feels" the "production line", at least to

a certain degree, and actively involves with it (e.g. bank loan). Thus it is safe to state that in services, the process is the product.

- human resource varies in knowledge/specification/abilities and the product is not standard, which result in difficulties in the process of standardizing the procedures.

In services like Informatics and Software Development, congestion usually becomes perceived only when employees complain about the workload. Waste is concealed in the processes, through which the data flow is made. And the employees' inactivity can easily be covered behind a PC. For the aforementioned reasons, detailed investigation of all of the procedures' steps is vital, in order to uncover the waste spots.

Lean Production, in comparison to mass production, utilizes less humans, space, machinery, time (procedures' improvement), stock. It also improves quality, along with productivity, reduces waiting periods, where this can be done, and releases productive resources.

Lean's core philosophy, both in production and services, is the same, the maximization of value which is requested (and paid for) by the customer, with simultaneous elimination of productive resources' waste.

We could say that Lean means:

- The belief that there is always a better way for a procedure/job to be executed
- Continual attempt of spotting and eliminating waste
- Reduction of processes' time
- Participation of all who are entangled in improvement of procedures as well as the working environment (including the workers (or programmers in SW) that actually do the job)
- Proactive functioning through mistake proof

As far as Lean implementation in Informatics is concerned and more specifically in Software Development, there is definitely fertile ground. Statistic evidence shows that a significant percentage of Informatics projects are cancelled, postponed, prone to mistakes, and fail to comply with the original specifications. All of those factors lead to increased costs, delays, stressing and uncomfortable workplace, and unsatisfied customers.



Bibliography, as well as my personal experience, signify the difficulty of the adaptation of IT community to Lean culture, for it opposes the current status quo. Lean philosophy demands discipline, vigilance, organization and not leaving things in the hands of fate.

In the Case Study of chapter 9 Lean's capability of implementation and its possible beneficial effects in a SAP Software Development department are examined. The current situation is described along with the problems that occur, and I make specific proposals about improvement of procedures and increment of value for the customer. My ideas are based on Kaizen (continual improvement) and in reduction of waste in the value stream, as I consider those two parts are actually the heart of Lean.

The current state in this specific department, as in most companies, is rough. There are many requests for change/improvement/projects that cause unrest and stress in the sector's employees who attempt to deal with all of them as fast as possible. In this effort of theirs it is rational that mistakes and omissions happen. In addition to that, due to the pressure, the time used for initial planning of solutions and identifying the real cause of the problems is inadequate, a fact which results mostly in firefighting (reactive, not proactive) actions. In this and all SW departments in general, the principal propositions for upturning the current state of facts, are:

- writing of **fewer** and **more quality** code, that will be transferred to production **without delay**.
- allocating more time in the planning section, so that the true cause of malfunctions and their solutions are identified and there are no repeating problems. Such an idea is training the users to avoid mistakes and not to ask for help often, or creation of validations that would prohibit faulty actions (mistake proof system).
- disallowance of leaving work semi-completed (that is changes that do not reach production system).
- distribution of knowledge created in personal level in the entire department (e.g. through appropriate updates of the company's portal).
- reduction of bureaucratic procedures of changes' approval that offer no true benefits.
- creation of appropriate KPIs that will support actual (lean) improvement of processes.

The most important conclusion is that Lean is not readily available for someone to mimic or copy. Every organization must discover how and where can Lean principles be implemented. Especially in the chaotic world of a Software Development department, steps of Lean implementation like those I suggest are able to produce visible advantages.

## ΚΕΦΑΛΑΙΟ 1. ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Το Lean σαν φιλοσοφία και πρακτική έχει τις ρίζες του στη μαζική παραγωγή. Ουσιαστικά εξελίσσει και βελτιώνει τη μαζική παραγωγή. Στην πορεία αυτής της εξέλιξης υπήρξαν κάμποιοι βασικοί σταθμοί οι οποίοι κρίνεται σκόπιμο να αναφερθούν.

### 1.1 Interchangeable parts

Η πρώτη σημαντική εφεύρεση που μπορεί να θεωρηθεί η βάση της μαζικής παραγωγής παρουσιάστηκε **τον Ιούλιο του 1785 στο Παρίσι**, από τον οπλοποιό **Honore Blanc**. Ο Blanc είχε καταφέρει να δημιουργήσει συναρμολογούμενα (εναλλάξιμα) εξαρτήματα όπλων για το γαλλικό στρατό. Στην επίδειξη που έκανε αποσυναρμολόγησε τους μηχανισμούς εκπυρσοκρότησης (locks) από 50 τυφέκια, έβαλε τα κομμάτια σε κουτιά και στη συνέχεια οι επισκέπτες συναρμολόγησαν γρήγορα και εύκολα τους μηχανισμούς λαμβάνοντας τυχαία κομμάτια και τους τοποθέτησαν ξανά στα τυφέκια.

Στη συγκεκριμένη επίδειξη βρισκόταν και ο πρεσβευτής των Ηνωμένων Πολιτειών στη Γαλλία, Thomas Jefferson. Ο μελλοντικός 3<sup>ος</sup> πρόεδρος των Ηνωμένων Πολιτειών διέβλεψε τη μεγάλη δύναμη της νέας εφεύρεσης. Ο Αμερικανικός πόλεμος για την ανεξαρτησία από τη Μ. Βρετανία είχε τελειώσει λίγο καιρό πριν (συνθήκη του Παρισιού [3 Σεπτεμβρίου 1783](#)), και ο στρατός των Ηνωμένων Πολιτειών αντιμετώπιζε ελλείψεις σε όπλα. Εάν μπορούσαν να παραχθούν εύκολα αυτά τα συναρμολογούμενα εξαρτήματα, θα μπορούσαν να συναρμολογηθούν τα τυφέκια επίσης εύκολα από **χαμηλόμισθους ανειδίκευτους εργάτες**. Έτσι δε θα χρειαζόνταν ειδικοί τεχνίτες για την κατασκευή των τυφεκίων, που σημαίνει μεγαλύτερη ταχύτητα κατασκευής και λιγότερα έξοδα.

Εδώ πρέπει να σημειωθεί η μεγάλη σημασία που είχε αυτή η εφεύρεση για την ανθρωπότητα. Αυτή η λογική της συναρμολόγησης και εναλλαξιμότητας των μερών, μπορεί να θεωρείται αυτονόητη στην εποχή μας. Εκείνη όμως την περίοδο ήταν **μία μεγάλη επανάσταση στον τρόπο σκέψης και βιομηχανικής κατασκευής**. Επίσης η παγκόσμια επέκτασή της δεν πραγματοποιήθηκε χωρίς δυσκολίες και αντιστάσεις. Στη Γαλλία υπήρξε ισχυρό κίνημα εναντίον της εφεύρεσης τόσο από τους οπλοποιούς όσο και από τους απλούς ανθρώπους, λόγω της επίφοβης απώλειας εργατικών θέσεων. Το κίνημα αυτό επεκτάθηκε και στην πολιτική.

Το αποτέλεσμα ήταν να επικρατήσουν οι αντίθετοι με την εφεύρεση και η Γαλλία να χάσει ακόμα και την τεχνογνωσία που είχε δημιουργήσει. Στην Αγγλία αντίστοιχα υπήρξαν έντονες διαμαρτυρίες εναντίον της εφεύρεσης μηχανών κλωστοϋφαντουργίας, πάλι εξαιτίας της πιθανής απώλειας εργατικών θέσεων. Όμως στις Ηνωμένες Πολιτείες που τα εργατικά χέρια έλειπαν και δεν υπήρχε έντονη και ριζωμένη εργατική χειρωνακτική κουλτούρα, η νέα ιδέα βρήκε γόνιμο έδαφος ώστε να εξαπλωθεί και να κατακτήσει τον κόσμο.

Ο Thomas Jefferson είχε μεταφέρει στην Αμερική εκτός από την ιδέα της εναλλαξιμότητας μερών (interchangeability), και δείγματα των συναρμολογούμενων τυφεκίων. Η προσπάθειά του για την υλοποίηση αυτής της νέας ριζοσπαστικής ιδέας, είχε ως αποτέλεσμα τη δημιουργία δύο εργοστασίων όπλων, στο Springfield στη Μασαχουσέτη και στο Harpers Ferry στη Δυτική Βιρτζίνια. Ήταν στο Harpers Ferry που κατάφεραν και δημιούργησαν τους πρώτους συναρμολογούμενους και εναλλάξιμους μηχανισμούς εκπυροκρότησης (locks).

Ο οπλοποιός και εφευρέτης που ασχολούνταν με την έρευνα και την υλοποίηση της νέας ιδέας εκεί ήταν ο **John Hancock Hall**, ο οποίος είχε να αντιμετωπίσει τεράστια και πολλά προβλήματα τόσο τεχνικά αλλά και διοικητικά. Επί της ουσίας και όσον αφορά το τεχνικό κομμάτι, διαπίστωσε ότι για να πετύχει η εναλλαξιμότητα των αποσπώμενων μερών, αυτό που απαιτείτο ήταν μεγάλη ακρίβεια κατασκευής των μερών. Ενώ τα συμβατικά μέρη ενός τυφεκίου είχαν ακρίβεια  $\pm 0.2$  mm, χρειαζόταν  $\pm 0.02$  mm για την επίτευξη της εναλλαξιμότητας. Για να πετύχει το σκοπό του, ο Hall εφηύρε νέες μεθόδους μέτρησης, δημιούργησε μηχανήματα μείωσης των κραδασμών και βελτίωσε διαδικασίες παραγωγής. Το αποτέλεσμα όλης αυτής της έρευνας και των προσπαθειών ήταν το **1824** τα τμήματα των μηχανισμών εκπυροκρότησης (locks), να είναι πλήρως εναλλάξιμα στο Harpers Ferry.

Το επόμενο βήμα ήταν αυτή η εναλλαξιμότητα να επεκταθεί και ανάμεσα στα δύο διαφορετικά εργοστάσια. Τη δουλειά αυτή την έκαναν ο Hall μαζί με τον οπλοποιό **Simeon North**. Το **1834** είχε επιτευχθεί και η εναλλαξιμότητα ανάμεσα στα δύο στρατιωτικά οπλοποιεία.

Ένα άλλο σημαντικό πρόσωπο στη νέα πορεία της Αμερικής προς τη μαζική παραγωγή, ήταν ο **Eli Whitney**. Ο Whitney ήταν εφευρέτης και επιχειρηματίας και είχε ενστερνιστεί την εναλλαξιμότητα των τμημάτων. Είχε εφεύρει τη μηχανή καθαρισμού του βαμβακιού (cotton gin), την οποία ήθελε να χρησιμοποιήσει ως τεχνολογία και σε άλλους επικερδείς

τομείς. Προχώρησε λοιπόν και στη δημιουργία εταιρείας όπλων, έχοντας κερδίσει ένα συμβόλαιο με την Αμερικανική κυβέρνηση για την παράδοση 10.000 τυφεκίων σε δύο χρόνια το 1798.

Είχε σχεδιάσει μηχανικά εργαλεία με τα οποία ένας ανειδίκευτος εργάτης μπορούσε να παράξει ένα συγκεκριμένο τμήμα του όπλου με την απαιτούμενη ακρίβεια. Τα διαφορετικά αυτά τμήματα συναρμολογούμενα δημιουργούσαν ένα τυφέκιο. Τελικά η παράδοση των τυφεκίων από τον Whitney καθυστέρησε πάνω από 10 χρόνια. Παρόλα αυτά είχε ήδη εγκαθιδρυθεί η έννοια της μαζικής παραγωγής εναλλάξιμων τμημάτων η οποία πλέον θα χρησιμοποιούνταν σε όλους τους τομείς της βιομηχανίας και θα άλλαζε τον κόσμο μας.

## 1.2 Interchangeable (Αναλώσιμοι) Εργάτες.

Στις αρχές του 20<sup>ου</sup> αιώνα τρεις σημαντικές προσωπικότητες της εποχής που ασχολήθηκαν με την βελτίωση των διαδικασιών στις βιομηχανικές εργασίες ήταν οι **Frederick Winslow Taylor** και οι **Frank** και **Lillian Gilbreth**.

Ο Taylor πίστευε ότι οι περισσότεροι μισθωτοί χειρωνακτες εργάτες σε γραμμές παραγωγής, προσπαθούσαν να δουλέψουν λιγότερο από αυτό που πραγματικά μπορούσαν, διότι δε θα κέρδιζαν κάτι παραπάνω με μεγαλύτερη απόδοση. Ο στόχος του ήταν να βρει τη βέλτιστη διαδικασία για τις χειρωνακτικές δουλειές που πραγματοποιούσαν οι εργάτες, να τηματοποιήσει τα βήματα της εργασίας στο μέγιστο βαθμό και να μετρήσει τους απαιτούμενους χρόνους. Έτσι ο κάθε εργάτης στη γραμμή παραγωγής θα έκανε τη δική του και μόνο εργασία στον ορισμένο χρόνο ώστε να μην διακόπτεται ή καθυστερεί η παραγωγή.

Αντίστοιχη ήταν και η έρευνα του Frank Gilbreth. Ο Gilbreth μελέτησε τις ανθρώπινες κινήσεις και τις διαίρεσε σε 18 μεμονωμένες κινήσεις, τις οποίες συνδύαζε ώστε να πετύχει γρηγορότερο και αποδοτικότερο αποτέλεσμα. Χαρακτηριστικό παράδειγμα είναι οι χτίστες, για τους οποίους είχε στήσει σκαλωσιά έτσι ώστε ο εργάτης, τα τούβλα και ο τοίχος να βρίσκονται όλα στο ίδιο ύψος και να μη χρειάζεται να λυγίζει ο εργαζόμενος.

Επίσης πολύ σημαντική είναι η συμβολή του Gilbreth στη λειτουργία των χειρουργείων, διότι ακολουθήθηκε η καθιερωμένη σε μας μέθοδος, τα εργαλεία να τα δίνει ο βοηθός νοσοκόμος και να μην τα φάχνει ο ίδιος ο χειρουργός. Με αυτήν την απλή (αλλά και πόσο σύνθετη!) σκέψη, αυξήθηκαν τα ποσοστά επιτυχίας των επεμβάσεων και επιβίωσης των ασθενών.

Όσον αφορά τη Lillian Gilbreth, συνεισέφερε και εκείνη στην έρευνα του συζύγου της και ήταν από τις πρώτες γυναίκες μηχανικούς και η πρώτη βιομηχανική ψυχολόγος. Στη Lillian Gilbreth οφείλεται ο σχεδιασμός της σύγχρονης κουζίνας, με πάγκο εργασίας γύρω γύρω στους τοίχους και ράφια/ντουλάπια από πάνω και από κάτω από τον πάγκο.

### 1.3 Μαζική παραγωγή αυτοκινήτων.

Στις Ηνωμένες Πολιτείες το 1914 **Henry Ford** αυξάνει τους μισθούς των εργατών του από **2,40\$ για 9 ώρες δουλειάς σε 5\$ για 8 ώρες**, στη γραμμή παραγωγής του μοντέλου T.

Η κίνησή του θεωρήθηκε υπερβολική και παράλογη, αλλά ο Ford ήταν πολύ έξυπνος και στόχευε στη δημιουργία μιας μεσαίας τάξης που θα είχε χρόνο και χρήμα να αγοράσει το αυτοκίνητο που εκείνος παρήγαγε και πουλούσε. Εφαρμόζοντας τις ιδέες του Taylor στο μοντέλο T, είχε καταφέρει να μειώσει το χρόνο συναρμολόγησης ενός αυτοκινήτου από **12 ώρες σε 90 λεπτά**.

Η κάθε δουλειά στη γραμμή συναρμολόγησης ήταν πολύ βαρετή, επαναλαμβανόμενη και απόλυτα ελεγχόμενη. Οι εργάτες μπορούσαν να εκπαιδευτούν μέσα σε 10 λεπτά, καθώς επίσης και να **αντικατασταθούν** σε 10 λεπτά. Αυτοί οι **«αντικαταστάσιμοι» «αναλώσιμοι»** εργάτες βρίσκονταν στο επίκεντρο ενός νέου βιομηχανικού μοντέλου : **της Μαζικής Παραγωγής**. Οι υψηλοί μισθοί εξυπηρετούσαν το σκοπό να απαλύνουν τη βαρετή και χωρίς ενδιαφέρον εργασιακή ζωή τους.

Το 1923 γίνεται CEO της **General Motors** ο **Alfred Sloan**. Εκείνη την περίοδο η General Motors κατείχε το 12% της αγοράς αυτοκινήτου. Την επόμενη χρονιά ο Sloan ανακοίνωσε ένα σχέδιο για «αυτοκίνητο για κάθε πορτοφόλι και σκοπό.» (*"car for every purse and purpose"*). Ο Sloan είχε καταλάβει ότι πλέον το αυτοκίνητο ήταν κάτι περισσότερο από μεταφορικό μέσο. Κατεδείκνυε το status και την προσωπική οπτική του αγοραστή του.

Το 1927 σταματάει η παραγωγή του μοντέλου T. Η μαζική παραγωγή του μοντέλου T για αρκετό καιρό ήταν προσοδοφόρα και ο Ford ήταν ο κυρίαρχος της αγοράς. Όμως η ανερχόμενη και ευκατάστατη μεσαία τάξη επιζητούσε την αλλαγή, τη διαφορετικότητα και την ποικιλία. Ο Ford δεν μπόρεσε να ανταποκριθεί γρήγορα στις απαιτήσεις των καιρών. Η γραμμή παραγωγής του δούλευε καλά μόνο για **ένα μοντέλο** και για **μεγάλο αριθμό αυτοκινήτων**. Στο μεταξύ η ανταγωνίστρια General Motors με τον Alfred Sloan ως CEO, είχε δημιουργήσει μια ευέλικτη εταιρεία που της επέτρεπε να παράγει πολλά διαφορετικά

μοντέλα τα οποία στόχευαν σε συγκεκριμένα τμήματα (segments) του καταναλωτικού κοινού και έως το 1927 είχε πουλήσει τα διπλάσια αυτοκίνητα από τη Ford.

#### 1.4 Το Σύστημα Παραγωγής Toyota (Toyota Production System).

Ο Sakichi Toyoda, στη δεκαετία του 1920, ήταν ο ιδιοκτήτης της εταιρείας Toyoda Automatic Loom Works. Ήταν κατασκευαστής αυτόματων αργαλειών στην Ιαπωνία. Έκανε συνεχή προσπάθεια για καλύτερο σχεδιασμό των αργαλειών, χρησιμοποιώντας τη γνώση εξειδικευμένων συνεργατών και με πνεύμα ετοιμότητας, αφουγκραζόμενος τις νέες τεχνολογικές εξελίξεις, όπως τα εναλλάξιμα εξαρτήματα. Προσελάμβανε ανθρώπους με γνώσεις τους οποίους εμπιστευόταν και βασιζόταν πάνω τους για την έρευνα και την υλοποίηση καινοτόμων ιδεών για την αυτόματη λειτουργία των αργαλειών.

Ο Kiichiro Toyoda, γιός του Sakichi, από τα κέρδη της επιχείρησης των αργαλειών, ξεκίνησε μια εταιρεία κατασκευής αυτοκινήτων, ως τμήμα της μητρικής Toyoda. Το 1937 όμως ιδρύθηκε η Toyota Motor Corporation ως ανεξάρτητη εταιρεία.

Το πρώτο αυτοκίνητο της Toyota βγήκε το 1936, αλλά σύντομα η παραγωγή διακόπηκε εξαιτίας του πολέμου.

Το 1945 που η Ιαπωνία βγαίνει από τον πόλεμο, είναι οικονομικά κατεστραμμένη, αλλά ο Kiichiro προτρέπει την εταιρεία του να φτάσει την Αμερικάνικη παραγωγή (*catch up with America*). Ήταν όμως ξεκάθαρο ότι αυτό δε θα μπορούσε να γίνει με τους όρους που λειτουργούσε η Αμερικανική παραγωγή, η οποία στηριζόταν στο μοντέλο της μαζικής παραγωγής. Αυτό σήμαινε την παραγωγή χιλιάδων ίδιων εξαρτημάτων ώστε να επιτευχθούν οικονομίες κλίμακος, αλλά τα υλικά ήταν δυσεύρετα, οι παραγγελίες λίγες και υπήρχε η καταναλωτική απαίτηση για διαφοροποίηση και ποικιλία των προϊόντων.

Ο Kiichiro είχε ένα όραμα, όλα τα εξαρτήματα για τη συναρμολόγηση να φτάνουν στη γραμμή συναρμολόγησης, ακριβώς τη στιγμή της χρήσης τους «*Just In Time*». Αυτό δεν έπρεπε να γίνει με αποθεματοποίηση, αλλά με την παραγωγή των εξαρτημάτων «*Just In Time*».

Το όραμα αυτό πήρε χρόνο για να γίνει πραγματικότητα, αλλά το 1962, μια δεκαετία μετά το θάνατο του Kiichiro, η εταιρεία του υιοθέτησε πλήρως το Toyota Production System.

### 1.4.1 Τα πρόσωπα πίσω από το TPS (Toyota Production System).

#### **Taiichi Ohno.**

Ήταν υπεύθυνος μηχανικός στην Toyota και ο «πατέρας» του TPS. Μοιραζόταν το όραμα του Kiichiro και το έκανε πράξη στην Toyota. Μελέτησε το σύστημα παραγωγής της Ford, όσο και τον τρόπο με τον οποίο τα αμερικάνικα supermarket χειρίζονταν τα αποθέματά τους. Παράλληλα αφουγκραζόταν και έδινε βάση στις παρατηρήσεις των υφισταμένων του εργατών. Χρειάστηκαν χρόνια πειραματισμών για τη δημιουργία και υλοποίηση του TPS, μια διαδικασία την οποία ο Ohno θεωρούσε συνεχιζόμενη (*never ending*). Στο βιβλίο του, **Toyota Production System**, αποκαλεί το TPS «ένα σύστημα για την απόλυτη εξάλειψη της σπατάλης». Εκεί εξηγεί ότι το σύστημα βασίζεται σε δύο πυλώνες: «*Just-in-Time flow*» και «*Autonomation*» (στα Ιαπωνικά Jidoka). Θα γίνει αναφορά στα συγκεκριμένα θέματα στο κεφ.2

#### **Shigeo Singo**

Ήταν σύμβουλος ο οποίος βοήθησε τον Ohno να υλοποιήσει το TPS στην Toyota, και στη συνέχεια βοήθησε κι άλλες εταιρείες να το κατανοήσουν και να το εφαρμόσουν.

Ο Singo στο βιβλίο του «*Study of Toyota Production System*», αναλύει δύο μεγάλα θέματα: την παραγωγή χωρίς απόθεμα (*nonstock production*) και τη μηδενική επιθεώρηση (*zero inspection*). Τα δύο αυτό θέματα ουσιαστικά αντικατοπτρίζουν τους δύο πυλώνες του Ohno.

#### **Eiji Toyoda**

Ξάδερφος του Kiichiro, ο οποίος κρατούσε καίρια διευθυντικά πόστα στην εταιρεία μετά το θάνατο του Kiichiro το 1950.

Και ο Eiji όπως και ο Kiichiro, είχε αντιληφθεί ότι το «παιχνίδι παίζεται» όχι στις οικονομίες κλίμακας, αλλά στο πώς θα διαχειριστεί κανείς την πολυπλοκότητα και την ποικιλία.

Υποστήριξε ενεργά τον Ohno στην επίτευξη του TPS, απέναντι στις τεράστιες αντιδράσεις που προκάλεσε η ροή “Just-in-Time”, διότι πήγαινε αντίθετα στο ρεύμα και την επικρατούσα λογική της μαζικής παραγωγής.



## 1.5 Ιαπωνική εξάπλωση.

Εκτός από την Toyota και άλλες Ιαπωνικές εταιρείες προσπαθούσαν να βγουν από την κρίση που είχε επέλθει λόγω του πολέμου. Γνωστοί επιστήμονες που βοήθησαν προς αυτήν την κατεύθυνση με τις μελέτες τους, τις παρουσιάσεις τους και την συμβουλευτική που παρείχαν στα στελέχη Ιαπωνικών εταιρειών, ήταν οι **William Edwards Deming**, ο **Kaoru Ishikawa** και ο **Joseph Juran**.

Ο Deming εστίαζε στο στατιστικό έλεγχο ποιότητας(Statistical Quality Control) ενώ οι Ishikawa και Juran στο Quality Management.

Τα αποτελέσματα των μελετών τους και οι προτάσεις τους αντικατοπτρίζουν τη lean σκέψη και φιλοσοφία. Βασικά παραδείγματα είναι ο κύκλος PDCA/PDSA (Plan-Do-Check/Study-Act) του Deming και το Fishbone diagram του Ishikawa.

Αξίζει να σημειωθεί ότι το 1960 ο Deming βραβεύτηκε από τον τότε Πρωθυπουργό της Ιαπωνίας Nobosuke Kishi, προς αναγνώριση της συνεισφοράς του στην αναγέννηση της Ιαπωνικής οικονομίας.

Το TPS αγνοούνταν από τον επιχειρηματικό κόσμο, ακόμα και στην Ιαπωνία, διότι οι εταιρείες αναπτύσσονταν γρήγορα και μπορούσαν να πουλάνε ό,τι έφτιαχναν. Η πετρελαϊκή κρίση όμως του 1973, προκάλεσε πολλά προβλήματα και ανέδειξε εταιρείες όπως η Toyota, η οποία κατάφερε να βγει γρήγορα από την κρίση. Έτσι το TPS έγινε αντικείμενο μελέτης για άλλες Ιαπωνικές εταιρείες που εφάρμοσαν στοιχεία του.

Μέσα σε μια δεκαετία Αμερική και Ευρώπη άρχισαν να νιώθουν τον έντονο ανταγωνισμό από τους Ιάπωνες.

Για τη διερεύνηση της επιτυχίας των ιαπωνικών εταιρειών πραγματοποιήθηκε μελέτη και έρευνα από τους J.Womack, Daniel Jones και Daniel Roos, αποτέλεσμα της οποίας είναι το γνωστό βιβλίο του 1990 « **The Machine that Changed the World**».

Το αποτέλεσμα της έρευνας των J.Womack, Daniel Jones και Daniel Roos, εισήγαγε τον όρο Lean, θέλοντας να περιγράψουν ουσιαστικά το TPS, και πώς οι εταιρείες θα μπορούσαν να κάνουν πιο ομαλά τη μετάβαση από τη μαζική παραγωγή στη lean. Χρησιμοποιήθηκε ο όρος Lean διότι κατά τη μελέτη της Toyota, διαπίστωσαν ότι η εταιρεία χρησιμοποιούσε λιγότερους πόρους σε σύγκριση με τη μαζική παραγωγή : λιγότερους ανθρώπους, λιγότερο χώρο, λιγότερα εργαλεία, λιγότερες κατασκευαστικές ώρες για το λανσάρισμα νέου προϊόντος. Από τότε έχουν συγγραφεί πολλά βιβλία, μελέτες και άρθρα,

με θέμα τη lean φιλοσοφία, πράγμα που αποδεικνύει ακόμα πιο έντονα τη σημασία της Lean σκέψης.

### 1.6 Lean και υπηρεσίες.

Η lean παραγωγή κυριαρχεί πλέον σε όλη την αυτοκινητοβιομηχανία. Η lean σκέψη έχει πλέον περάσει και σε άλλους τομείς όπως στις υπηρεσίες, τα logistics, φαρμακοβιομηχανίες product και software development. Οι εταιρείες που ασχολούνται με ανάπτυξη νέων προϊόντων αλλά και software, εάν θέλουν να επιβιώσουν και να αναπτυχθούν οφείλουν να ακολουθήσουν τις επιταγές της lean σκέψης. Αυτό δεν είναι πάντα εύκολο. Χρειάζεται αλλαγή τόσο στην κουλτούρα των ανθρώπων όσο και της ίδιας της εταιρείας. Είναι σίγουρο όμως πως τα οφέλη είναι ανυπολόγιστα από την ορθή εφαρμογή του lean.

Ειδικά για το SW Development υπήρξε μια αξιόλογη προσπάθεια το 2001 από 17 προγραμματιστές που έθεσαν τις βασικές αρχές που πρέπει να διέπουν την ανάπτυξη κώδικα, ώστε να δημιουργείται ποιοτικό software το οποίο θα αυξάνει την αξία για τον πελάτη, θα παραδίδεται ταχύτερα, παράλληλα όμως θα μειώνονται και τα κόστη. Πρόκειται για το **«Manifesto for Agile Software Development»**. Οι 12 αρχές του μανιφέστου έχουν τις ρίζες τους στη Lean φιλοσοφία και αντιμετωπίζουν τη δημιουργία SW ως δημιουργία νέου προϊόντος (**product development**) και όχι ως συνεχή κατασκευή ίδιων προϊόντων (manufacturing). Αυτή η προσπάθεια δείχνει πόσο μεγάλη επίδραση έχει και στο χώρο του SW η Lean σκέψη.

Στο επόμενο κεφάλαιο θα αναφερθούμε στις βασικές αρχές της lean φιλοσοφίας εστιάζοντας ιδιαίτερα στο IT και στο SW development.

## ΚΕΦΑΛΑΙΟ 2 . ΤΙ ΕΙΝΑΙ LEAN

### 2.1 Η έννοια του Lean.

Από τη βιβλιογραφία και την εμπειρία γίνεται αντιληπτό ότι η έννοια και η εφαρμογή του Lean στους οργανισμούς δεν είναι απόλυτα ξεκάθαρη. Πολλές φορές στην προσπάθεια των εταιρειών να ακολουθήσουν τη Lean φιλοσοφία που έχει αποδειχτεί αποτελεσματική σε κάποια άλλη εταιρεία, προσπαθούν απλά να μιμηθούν την επιτυχημένη εταιρεία όπως το αντιλαμβάνονται, με απογοητευτικά αποτελέσματα. Στη συνέχεια θα γίνει προσπάθεια να ξεκαθαρίσει το τοπίο σχετικά με το Τι είναι και Τι δεν είναι η Lean φιλοσοφία.

Η έννοια της Lean φιλοσοφίας είναι μια εμπειρική, και για το λόγο αυτό μοναδική, διαδικασία πειραματισμού και μάθησης. Σκοπός είναι να εφαρμοστεί η φιλοσοφία αυτή σε κάθε διαφορετικό οργανισμό με τρόπους που ταιριάζουν στο συγκεκριμένο οργανισμό. Ο στόχος είναι πάντα η δημιουργία της μέγιστης αξίας για τον πελάτη με το μικρότερο κόστος. Το Lean δεν είναι κάποιο συγκεκριμένοι κανόνες ή μέθοδοι οι οποίες μπορούν εύκολα να εφαρμοστούν οπουδήποτε και να έχουν το επιθυμητό αποτέλεσμα. Δεν είναι ένα σύνολο οδηγιών.

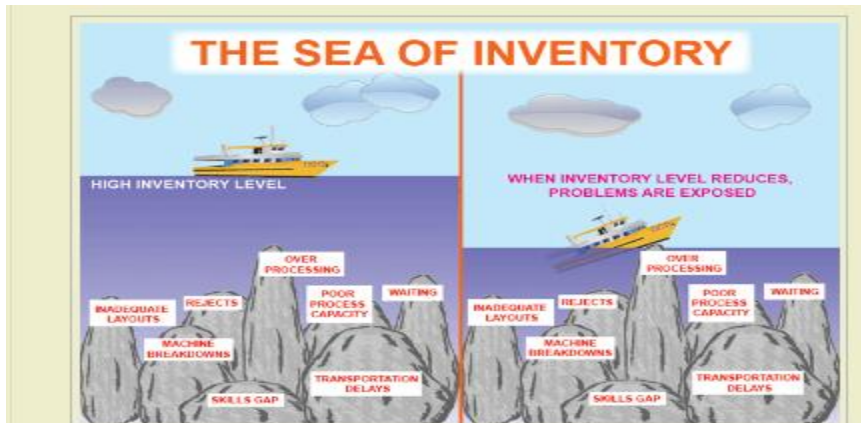
Το Lean αφορά περισσότερο την κουλτούρα και τον τρόπο σκέψης του ανθρώπου και κατ' επέκταση της εταιρείας με σκοπό :

- Τη βελτίωση της ποιότητας
- Τη μείωση του κόστους
- Την ταχύτερη παράδοση
- Την παραγωγικότητα
- Την κερδοφορία
- και γενικά τη **συνεχή βελτίωση της συνολικής απόδοσης του οργανισμού.**

Η λιτή(lean) παραγωγή σε σχέση με τη μαζική, χρησιμοποιεί λιγότερους ανθρώπους, χώρους, μηχανήματα, χρόνο (βελτίωση διαδικασιών), αποθέματα. Βελτιώνει επίσης την παραγωγικότητα και την ποιότητα, μειώνοντας τους χρόνους αναμονής (οπουδήποτε μπορεί αυτό να γίνει) και αποδεδειγμένα παραγωγικούς πόρους.

Η βασική φιλοσοφία είναι η μεγιστοποίηση της αξίας που λαμβάνει(πληρώνει) ο πελάτης, με ταυτόχρονη εξάλειψη της σπατάλης παραγωγικών πόρων, όταν δε δημιουργείται αξία για τον τελικό πελάτη.

Ένα βασικό παράδειγμα σπατάλης είναι η υπερ-παραγωγή και η υπερ-αποθεματοποίηση.



Σχήμα 1. Η θάλασσα των αποθεμάτων

Τα πολλά αποθέματα κρύβουν προβλήματα όπως ελαττωματικά προϊόντα, προϊόντα που δε θα πουληθούν ποτέ διότι δεν τα ζήτησε ο πελάτης, ανενεργό (χαμένο) χρόνο για ελέγχους αποθεμάτων, κόστος αποθεματοποίησης και διατήρησης του αποθέματος κλπ. Το Lean απαιτεί να έρχονται στην επιφάνεια και να λύνονται ριζικά τα προβλήματα που υπάρχουν και όχι να «κρύβονται» π.χ. κάτω από τη θάλασσα του αποθέματος.

Η Lean φιλοσοφία όπως γνωρίζουμε, έχει τη βάση της στο Toyota Production System. Η Toyota στη δεκαετία του 1960 προσέφερε στους πελάτες της μεγαλύτερη ποικιλία υψηλής ποιότητας αυτοκινήτων χωρίς να είναι υπερβολικά ακριβά. Η γενικότερη απαίτηση των καταναλωτών για ποικιλία μπόρεσε να ικανοποιηθεί απόλυτα και έγκαιρα από την Toyota, ενώ βρήκε απροετοίμαστες τις άλλες αυτοκινητοβιομηχανίες που χρησιμοποιούσαν συστήματα μαζικής παραγωγής. Η παραγωγή μεγάλης ποικιλίας ποιοτικών προϊόντων, σε μικρότερες ποσότητες απαιτούσε διαφορετική προσέγγιση με εστίαση πλέον στις οικονομίες ροής (economies of flow) και όχι κλίμακας (economies of scale). Η οικονομία ροής σημαίνει ότι το κέρδος θα προέλθει από την ταχύτερη παραγωγή και πώληση διατηρώντας το ελάχιστο δυνατό απόθεμα.

Θα μπορούσαμε να πούμε επιγραμματικά ότι lean σημαίνει:

- Την πεποίθηση ότι υπάρχει πάντα ένας καλύτερος τρόπος για μια συγκεκριμένη εργασία/διαδικασία
- Τη συνεχή προσπάθεια εύρεσης και εξάλειψης της σπατάλης
- Τη μείωση του χρόνου των διαδικασιών

- Τη συμμετοχή όλων των εμπλεκομένων στη βελτίωση των διαδικασιών και του εργασιακού περιβάλλοντος (εννοώντας και τους εργάτες που πραγματικά κάνουν τη δουλειά).

## 2.2 Just-In-Time / Autonomation(Jidoka).

Όπως είχε αναφερθεί στο προηγούμενο κεφάλαιο, το Toyota Production System είχε ως βασικούς πυλώνες τα «Just In Time flow» και το «Autonomation». Σε αυτούς τους πυλώνες βασίζεται και η Lean φιλοσοφία.

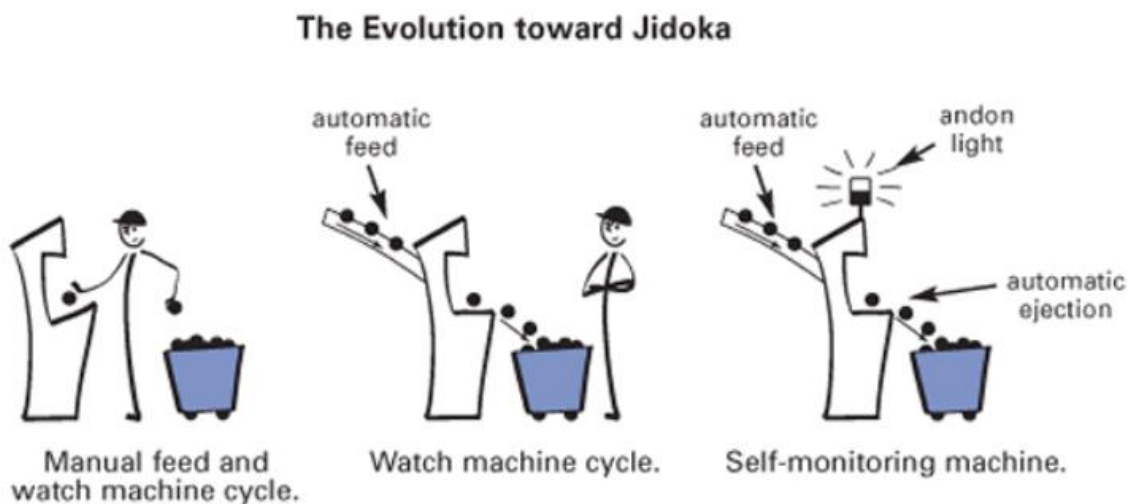
Η ροή **Just-in-Time** στοχεύει στην εξάλειψη του αποθέματος «**Nonstock Production**». Η λογική της αποθεματοποίησης χρησιμοποιείται για την επίτευξη οικονομίας κλίμακας, ενώ αντίθετα η λογική του Just-In-Time σκοπεύει σε παραγωγή πολλών, μικρών και ποικίλων παρτίδων που δε θα δημιουργήσουν απόθεμα.

Η έννοια της αυτονομίας **Autonomation(Jidoka)**, σημαίνει ότι η εργασία είναι έτσι οργανωμένη ώστε και η παραμικρή ανωμαλία γίνεται αμέσως αντιληπτή, η παραγωγή σταματάει, και συνεχίζεται μόνο εφόσον αποκατασταθεί η πραγματική αιτία του προβλήματος. Μια άλλη ονομασία αυτής της κρίσιμης λειτουργικότητας και πιο εύκολης στη μνήμη είναι : «**Stop-the-line**». Ο Ohno αποκαλούσε το “Autonomation” “*automation with a human touch*”, δηλ. αυτοματισμό με ανθρώπινη αίσθηση. Παραλλήλιζε ένα τέτοιο εργασιακό σύστημα με του ανθρώπινου οργανισμού. Ο άνθρωπος έχει το αυτόνομο νευρικό σύστημα που ρυθμίζει αναπνοή, καρδιακό ρυθμό, πέψη και ανατακλαστικά. Όταν κάποιος ακουμπήσει κάτι καυτό, το αυτόνομο νευρικό σύστημα λειτουργεί άμεσα και τον κάνει να το αφήσει, χωρίς να περιμένει το σήμα από τον εγκέφαλο. Έτσι κι ένας τέτοιος οργανισμός έχει ανατακλαστικά που του επιτρέπουν να ανταποκριθεί άμεσα και ορθά σε έκτακτα συμβάντα χωρίς να χρειαστεί να ζητήσει οδηγίες.( T. Ohno, 1978)

Είναι σίγουρο πως εάν υπάρχει η παραμικρή πιθανότητα να συμβεί ένα λάθος, αυτό **σίγουρα** θα συμβεί. Άρα πρέπει να γίνει το παν ώστε να προληφθεί το λάθος από τον αρχικό σχεδιασμό του συστήματος.

Η υλοποίηση της «αυτονομίας», σημαίνει να είναι έτσι σχεδιασμένο το σύστημα ώστε να μην επιτρέπει τα λάθη (**mistake proof**). Το σύστημα θα πρέπει να λειτουργεί χωρίς συστηματική επιθεώρηση (**Zero Inspection**). Δε θα πρέπει να υπάρχει συνεχώς κάποιος που θα ελέγχει τη λειτουργία των μηχανών ή την ποιότητα κάθε προϊόντος. Ένα απλό αλλά σαφές παράδειγμα mistake-proof είναι το καλώδιο video που ταιριάζει μόνο σε συγκεκριμένη θύρα του υπολογιστή, **χωρίς να επιτρέπει στο χρήστη να κάνει λάθος**.

Στο παρακάτω σχήμα φαίνεται πώς στο στάδιο του autonotation, το μηχάνημα «αναγνωρίζει» το ελαττωματικό προϊόν, σταματάει την παραγωγή και εμφανίζει κατάλληλο οπτικό σήμα(**andon light**) σε πίνακα ελέγχου, ώστε άμεσα να επιληφθεί ο υπεύθυνος.



Σχήμα 2. Autonotation (Jidoka)

Ο lean οργανισμός έχει ως στόχο τη μέγιστη δυνατή αξία προς τον πελάτη με τους λιγότερους δυνατούς πόρους. Για την επίτευξη αυτού του στόχου, πρέπει να αλλάξει η κουλτούρα του οργανισμού και να γίνεται προσπάθεια **βελτιστοποίησης της συνολικής ροής** των προϊόντων/υπηρεσιών, αντί της **επιμέρους** βελτιστοποίησης τεχνολογιών, υποδομών, τμημάτων.

### 2.3 Βασικές αρχές Lean σκέψης.

Στη συνέχεια παρουσιάζονται οι 5 βασικές αρχές που διέπουν τη Lean σκέψη, σύμφωνα με τους Womack και Jones (2003).



Σχήμα 3. Οι 5 Lean αρχές

- **Identify Value.** Ο προσδιορισμός της Αξίας, όπως την αντιλαμβάνεται ο πελάτης. Η αξία δημιουργείται από τον προμηθευτή, αλλά ο τελικός πελάτης είναι εκείνος που θα καθορίσει την αξία του αγαθού που αγοράζει, υλικού ή υπηρεσίας. Η έννοια της αξίας αφορά στο πόσο αισθάνεται ο πελάτης ότι αξίζει το αγαθό. Ο πελάτης ζητάει αγαθά και υπηρεσίες που θα ικανοποιήσουν την ανάγκη του, χωρίς περιττές διαδικασίες, καθυστερήσεις, ελαττώματα κλπ.
- **Map the Value Stream.** Αποτύπωση της αλυσίδας Αξίας.  
Είναι όλα τα βήματα και οι ενέργειες που απαιτούνται, από την αγορά των Α υλών έως την τελική παράδοση του προϊόντος στον πελάτη ή από την παραγγελία της υπηρεσίας έως την ανάλωσή της. Για να επιτευχθεί η εξάλειψη της σπατάλης (ο βασικός στόχος του Lean), θα πρέπει να είναι απόλυτα ακριβής και πλήρης η κατανόηση του value stream. Οι διαδικασίες πρέπει να ελέγχονται για το είδος της αξίας που προσθέτουν. Οτιδήποτε περιττό που δεν προσθέτει αξία για τον πελάτη πρέπει να καταργείται. Σύμφωνα με τους Womack και Jones (2005), η αποτύπωση του value stream σχεδόν πάντα καταλήγει στους παρακάτω τρεις τύπους ενεργειών:
  - Αυτές που σίγουρα προσθέτουν αξία
  - Αυτές που δεν προσθέτουν αξία αλλά είναι απαραίτητες
  - Αυτές που είναι απολύτως περιττή σπατάλη (**muda** στα ιαπωνικά)

- **Flow.** Η ροή. Στη lean φιλοσοφία τα βήματα στη value stream πρέπει να εκτελούνται οριζόντια, απρόσκοπτα και η ροή να είναι συνεχής. Εάν σε κάποιο σημείο η ροή σταματήσει και περιμένει, αυτή η αναμονή είναι σπατάλη. Αυτή η απαίτηση του lean για συνεχή ροή, απαιτεί ακόμα μεγαλύτερη αλλαγή στη νοοτροπία όλων όσων εμπλέκονται σε αυτήν, π.χ. στη δημιουργία ενός προϊόντος. Πρέπει όλη η δουλειά να οργανώνεται **βάσει του προϊόντος** που δημιουργείται και να 'κυλάει' χωρίς αναμονές και χωρίς λειτουργικά εμπόδια. Το αποτέλεσμα της υλοποίησης συνεχούς ροής στην παραγωγή επιφέρει μεγάλες μειώσεις στους χρόνους, σε σχέση με τη Batch διαδικασία.

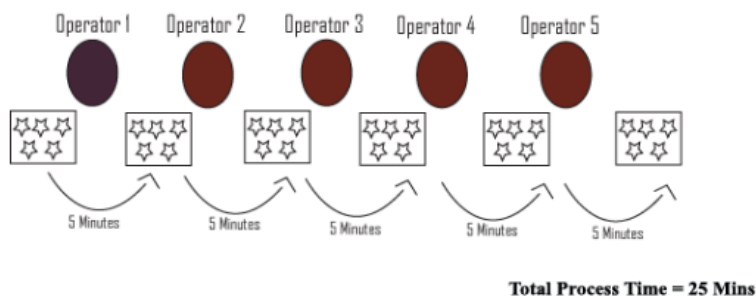
Αυτό μπορεί να γίνει πιο κατανοητό με το παρακάτω παράδειγμα:

Υποθέτουμε πως κάθε λειτουργία απαιτεί 1 min. Υπάρχει μια παρτίδα από 5 κομμάτια που πρέπει να επεξεργαστούν από τους αρμόδιους υπαλλήλους.

Ο πρώτος ξεκινάει με την παρτίδα των 5 κομματιών, για τα οποία κάνει 5 min.

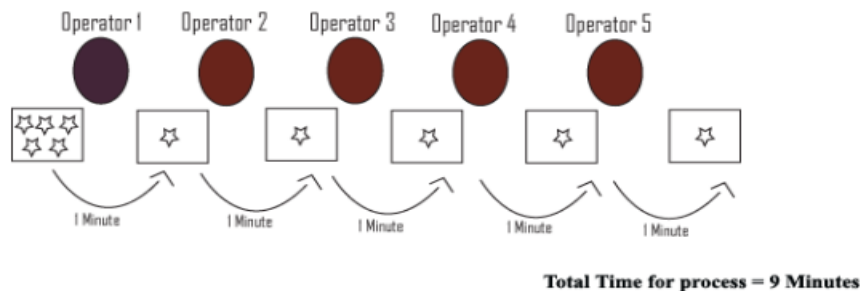
Όλη η παρτίδα αφού επεξεργαστεί προχωράει στον επόμενο κλπ. Συνολικός χρόνος 25 Min.

#### Traditional Batching Method - Example



Το ίδιο παράδειγμα, με συνεχή παραγωγή προσανατολισμένη στο προϊόν, συνολικός χρόνος 9 Min:

#### Lean Flow - Example



Σχήμα 4. Παράδειγμα Lean flow



- **Pull** σημαίνει ότι τα προϊόντα δημιουργούνται μετά από την απαίτηση του πελάτη και όχι ως πρόβλεψη προϊόντων, τα οποία πρέπει στη συνέχεια να πουληθούν (push). Έτσι δε δημιουργούνται αποθέματα, και ουσιαστικά τα προϊόντα έχουν πουληθεί πριν παραχθούν. Αυτή η λογική έχει να κάνει άμεσα με το Just-In-Time. Η αλήθεια είναι ότι η υλοποίησή της απαιτεί εξαιρετικό συντονισμό και επικοινωνία σε κάθε βήμα της αλυσίδας αξίας.
- **Perfection.** Τελειότητα. Η τελευταία αλλά θεμελιώδης αρχή της Lean φιλοσοφίας, είναι η συνεχής αναζήτηση της τελειότητας. Με τη λέξη τελειότητα δεν εννοείται η ποιότητα του προϊόντος, αλλά η εξάλειψη της σπατάλης και η συνεχής βελτιστοποίηση των διαδικασιών, των ανθρώπων και όλου του οργανισμού. Αυτή η διαδικασία δεν σταματάει ποτέ, όπως έλεγε και ο Όηνο. Θα πρέπει να είναι η κινητήρια δύναμη και το όραμα της εταιρείας, ώστε να εξελίσσεται συνεχώς και να αποκτάει ανταγωνιστικό πλεονέκτημα σε σχέση με τους ανταγωνιστές.

#### 2.4 Βασικά Lean «εργαλεία».

Το TPS στην πορεία της εξέλιξής του έχει δημιουργήσει βασικά εργαλεία μέσω των οποίων μπορεί να υλοποιηθεί το Lean. Εδώ πρέπει να τονιστεί ότι τα «εργαλεία» εξελίσσονται με την πάροδο του χρόνου και η εφαρμογή τους είναι τόσο προαιρετική όσο και προσαρμόσιμη σε κάθε περίπτωση. Κάποιος οργανισμός μπορεί να δημιουργήσει τα δικά του εργαλεία ή πρακτικές ώστε να επιτύχει τους Lean στόχους του. Αυτό που έχει σημασία είναι η εμπέδωση της Lean φιλοσοφίας από τον οργανισμό και η προσπάθεια επίτευξης των στόχων.

Θα γίνει αναφορά στα πιο γνωστά Lean εργαλεία του TPS που σε γενικές γραμμές ή με τις κατάλληλες προσαρμογές, μπορούν να χρησιμοποιηθούν σε όλους τους εργασιακούς χώρους.:

- **Andon.** Είναι visual management εργαλείο του Autonomation (Jidoka). Όταν ανιχνευτεί πρόβλημα σε κάποιο σταθμό εργασίας, ανάβει κατάλληλο σήμα σε πίνακα ελέγχου, ώστε να γίνουν οι απαραίτητες ενέργειες αποκατάστασης.
- **A3 Thinking.** Στην Toyota είναι μια management πρακτική και σημαίνει πώς ό,τι θέλουμε να παρουσιάσουμε για ένα πρόβλημα πρέπει να χωράει σε ένα χαρτί μεγέθους A3. Η περιγραφή αυτή συνήθως είναι σχηματική και

περιλαμβάνει την περιγραφή και επίλυση του προβλήματος, τους στόχους, την ανάλυση και το πλάνο δράσης.

- **5 S.** Αφορούν σε 5 βασικές δραστηριότητες για το χώρο εργασίας, που στα Ιαπωνικά ξεκινούν από S. Βέβαια τυχαίνει και στα Αγγλικά να ξεκινούν από S, κι έτσι είναι εύκολη η ανάκλησή τους στη μνήμη.

Ιαπωνικά	Αγγλικά	Σημασία	Παράδειγμα
<b>Seiri</b>	Sort	<b>Οργάνωση</b>	Ξεκαθάρισμα/πέταμα όλων των άχρηστων από το χώρο εργασίας
<b>Seiton</b>	Set in Order	<b>Τάξη</b>	Κάθε πράγμα στη θέση του, για εύκολη ανεύρεση και αποθήκευση
<b>Seiso</b>	Shine	<b>Καθαριότητα</b>	Ο χώρος εργασίας πρέπει να είναι απολύτως καθαρός (να αστράφτει)
<b>Seiketsu</b>	Standardize	<b>Τυποποίηση</b>	Να υπάρχει συγκεκριμένη διαδικασία και πρόγραμμα για τα παραπάνω
<b>Shitsuke</b>	Sustain	<b>Πειθαρχία/Δέσμευση</b>	Εμπέδωση της κουλτούρας των 5 S. Δεν τα κάνουμε μια φορά και τα ξεχνάμε στη συνέχεια.

Πίνακας 1. 5 S

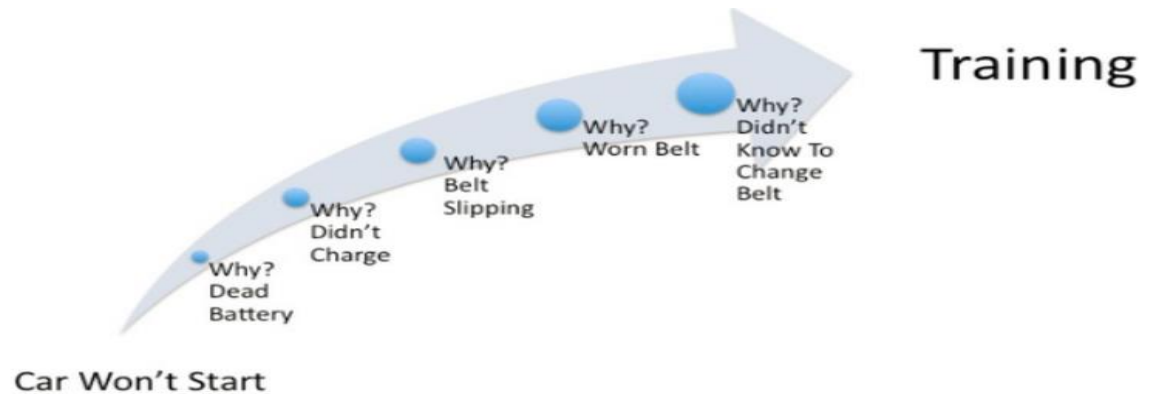
Μπορούν και πρέπει, να εφαρμόζονται σε όλους τους εργασιακούς χώρους, στο εργοστάσιο, στο γραφείο, στις δημόσιες υπηρεσίες και φυσικά στο σπίτι. Εφαρμόζονται εξίσου και σε υπολογιστικά περιβάλλοντα. Η τάξη και η οργάνωση π.χ. πρέπει να υπάρχουν και στα Folders του PC μας, ώστε να είναι ευχερέστερη η ανεύρεση και η αρχειοθέτηση. Παρόλο που αυτές οι απλές αρχές καθαριότητας και τάξης φαντάζουν αυτονόητες, στην πραγματικότητα δεν είναι και απαιτείται μεγάλη προσπάθεια για να γίνουν κήμα του καθένα και ακόμα περισσότερο για να ενσωματωθούν στην κουλτούρα της εταιρείας και του εργαζόμενου.

**Heijunka.** Σημαίνει εξισορρόπηση του φόρτου εργασίας. Στόχος είναι η πλήρης και ταχεία ικανοποίηση του πελάτη, χωρίς να υπερφορτώνεται ο εργαζόμενος. Σημειώνουμε ότι είναι ιδιαίτερα δύσκολο στο χώρο του IT και απαιτεί μεγάλη προσπάθεια.

**Poka Yoke.** Mistake-proofing. Είναι κάθε μηχανισμός αποφυγής λαθών από το χρήστη. Ένα τέτοιο παράδειγμα είναι τα καλώδια που συνδέονται σε διαφορετικές θύρες στο PC. Έχουν διαφορετικά μεγέθη ώστε να είναι ξεκάθαρο

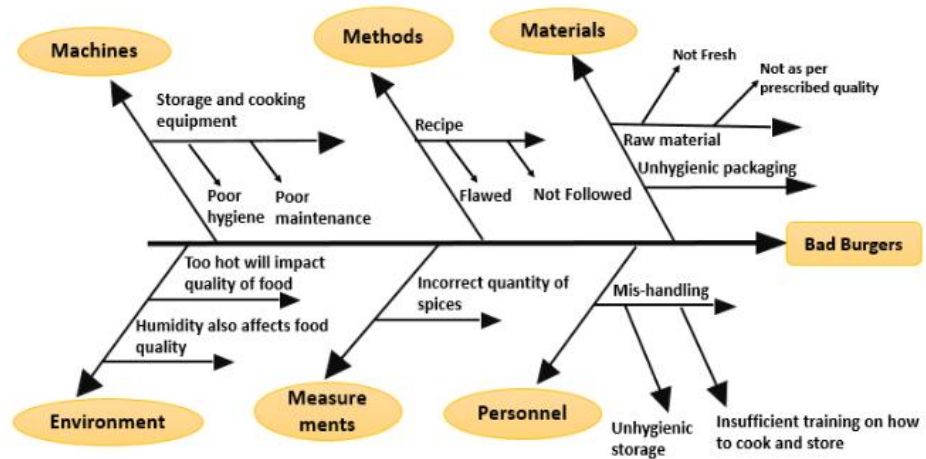
που μπαίνει τι αλλά και να αποτρέπεται το λάθος διότι το ένα δεν ταιριάζει στη θέση του άλλου.

**Why-Why Analysis.** Χρησιμοποιείται ώστε να φτάσει η ανάλυση στην πραγματική αιτία, στη ρίζα του προβλήματος. Λέγεται επίσης και 5 Whys. Κάποιες φορές πιθανά να χρειαστούν λιγότερες ή περισσότερες ερωτήσεις από 5 για να φτάσουμε στην αιτία.



Σχήμα 5. Why-Why analysis

**Fishbone Chart-Ishikawa diagram.** Επειδή η Why-Why ανάλυση δεν επαρκεί για επιχειρησιακά προβλήματα μεγάλου βεληνεκούς, σε αυτές τις περιπτώσεις προτιμώνται τα Fishbone Charts που είναι εργαλεία Cause and Effect Analysis. Αναπτύχθηκαν δε από τον Kaoru Ishikawa στη δεκαετία του 1960. Επιτυγχάνουν την οπτικοποίηση του προβλήματος και όλων των πιθανών αιτιών. Η κατηγοριοποίηση των αιτιών γίνεται σε : Manpower(people/staff), Materials, Methods (processes), Machines (equipment), Measurement και Environment.



Σχήμα 6. Fishbone diagram

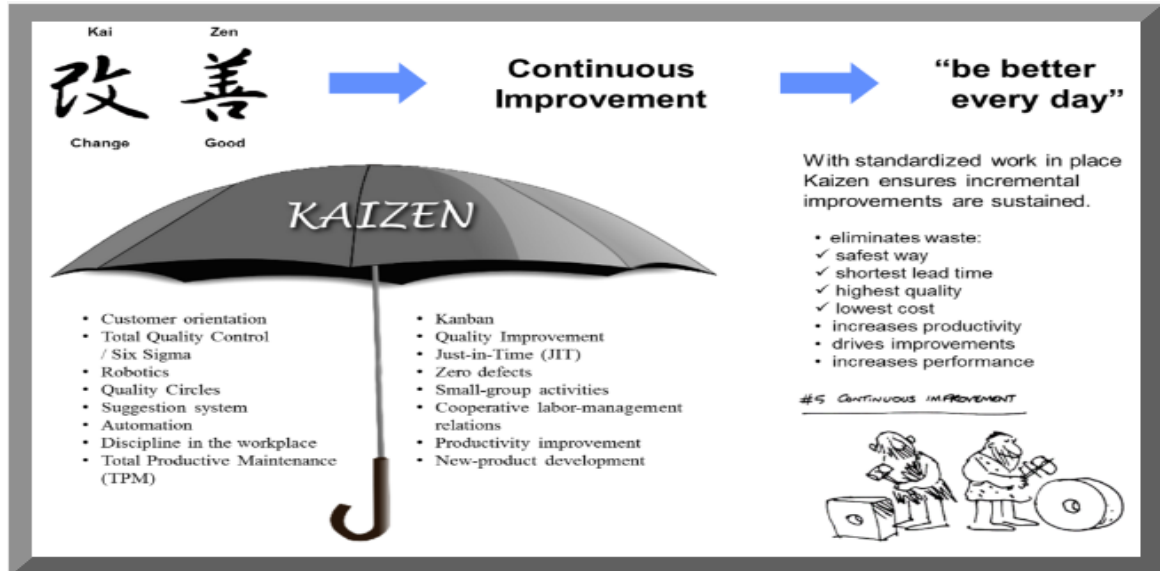
**PDCA**(Plan Do Check Act ). Είναι ο λεγόμενος κύκλος του Deming και χρησιμοποιείται τόσο για την επίλυση προβλημάτων όσο και για τη βελτίωση διαδικασιών. Αντίστοιχη είναι και η μέθοδος **DMAIC**( Define Measure Analyze Improve Control).

Παρακάτω οι βασικές μεθοδολογίες επίλυσης προβλημάτων:

Μεθοδολογίες επίλυσης προβλημάτων/βελτίωσης διαδικασιών		
PDCA	DMAIC	A3 thinking
<b>PLAN</b>	Define	1.Πλήρης αποσαφήνιση του προβλήματος
	Measure	2.Τμηματοποίηση του προβλήματος(breakdown)
		3.Προδιαγραφή στόχων
<b>DO</b>	Analyze	4.Ανάλυση/Εύρεση της αιτίας
		5.Ανάπτυξη της λύσης
	Improve	6.Εφαρμογή της λύσης
<b>CHECK</b>	Control	7.Αξιολόγηση αποτελεσμάτων και διαδικασιών, σύγκριση με προηγούμενη κατάσταση (benchmark
<b>ACT</b>		8.Τυποποίηση/επέκταση σε όλη την εταιρεία

Πίνακας 2. Μεθοδολογίες Επίλυσης Προβλημάτων

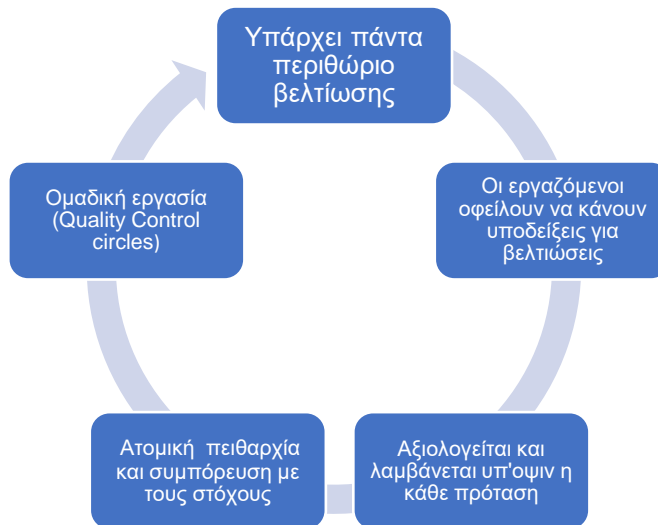
**Kaizen.** Είναι η κουλτούρα της συνεχούς βελτίωσης. Ίσως το σημαντικότερο εργαλείο, η «καρδιά» του Lean.



Σχήμα 7. Kaizen

Πραγματοποιείται με την επίλυση των προβλημάτων που ανιχνεύονται στο value stream mapping. Οι ομάδες έργου καταργούν τις δραστηριότητες που δεν παράγουν αξία. Κάτω από την «ομπρέλα» του Kaizen υπάρχουν οι μεθοδολογίες και οι έννοιες του Lean, ώστε να υλοποιηθούν οι στόχοι.

Παρακάτω παρουσιάζονται οι βασικές αρχές του Kaizen:



Σχήμα 8. Οι 5 βασικές αρχές του Kaizen

Όπως βλέπουμε στο σχήμα, στην κουλτούρα του Kaizen και του Lean παίζει τεράστιο ρόλο ο άνθρωπος, εργαζόμενος. Όχι μόνο ο manager αλλά και ο απλός εργαζόμενος που δουλεύει π.χ. στη γραμμή παραγωγής ή ο προγραμματιστής στο software development. Η συνεχής βελτίωση αφορά τον καθένα προσωπικά και την εταιρεία συνολικά. Ο εργαζόμενος έχει τη θέληση αλλά και τη δυνατότητα να κάνει προτάσεις βελτίωσης. Σημειώνεται ότι στην Toyota οι εργαζόμενοι υποβάλλουν περισσότερες από 10 υποδείξεις το χρόνο, και πάνω από το 99% αυτών υλοποιείται. Όπως λέει ο Fujio Cho (Επίτιμος Πρόεδρος της Toyota) :

*“We get brilliant results from average people managing brilliant processes, while our competitors get average or worse results from brilliant people managing broken processes.”*

Στον παρακάτω πίνακα βλέπουμε πώς χτίζεται η κουλτούρα της συνεχούς βελτίωσης:

1	Όλες οι διαδικασίες μπορούν και οφείλουν να βελτιώνονται
2	Δεν πρέπει να περνάει ημέρα χωρίς κάποια βελτίωση
3	Οραματιζόμαστε την ιδανική εμπειρία για τον πελάτη και προσπαθούμε να την υλοποιήσουμε
4	Δε μένουμε στην κριτική, αλλά προτείνουμε λύσεις
5	Σκεφτόμαστε πώς θα βελτιώσουμε κάτι και όχι γιατί δεν μπορεί να βελτιωθεί
6	Σκεφτόμαστε πέρα από την κοινή λογική. Ακόμα κι αν κάτι δουλεύει προσπαθούμε για την περαιτέρω βελτίωσή του
7	Η επίλυση των προβλημάτων πρέπει να είναι αποτέλεσμα συνεργασίας των εμπλεκόμενων τμημάτων / λειτουργιών

Πίνακας 3. Η κουλτούρα του Kaizen

Αυτή η αναζήτηση της συνεχούς βελτίωσης έρχεται σε αντίθεση με τη φυσική ροπή του ανθρώπου προς τη συνήθεια. Στους περισσότερους ανθρώπους είναι αρεστή η αλλαγή, αρκεί να μην αλλάζουν οι ίδιοι. Ειδικά όμως στο χώρο των επιχειρήσεων

θα πρέπει να είναι σαφές σε όλους τους εργαζόμενους, ότι η μη βελτίωση οδηγεί σε κατάρρευση. Η φιλοσοφία αυτή θα πρέπει να πηγάζει από την κορυφή της διοίκησης και να γίνει κουλτούρα του κάθε εργαζόμενου. Στο βιβλίο του *The Fifth Discipline*(1990), ο Peter Senge χρησιμοποιεί τον όρο *Learning Organization* θέλοντας να περιγράψει τον οργανισμό στον οποίο ο κάθε εργαζόμενος αντιλαμβάνεται την πολυπλοκότητα που υπάρχει, βλέπει το «όλον», το σκοπό και τους στόχους της εταιρείας και προσπαθεί να βελτιώνει καθημερινά τον εαυτό του, την εργασία του, τη συνεργασία με συναδέλφους και τμήματα ώστε να συνεισφέρει στο όραμα της εταιρείας. Ειδικά στην εποχή μας είναι ζήτημα βιωσιμότητας για τις εταιρείες να εξελιχθούν σε *Learning Organizations*, με άλλα λόγια να πραγματοποιούν το *Kaizen*.

## 2.5. Κριτικές για το Lean.

### 2.5.1 Απώλεια θέσεων.

Όπως έχει αναφερθεί ο βασικός στόχος του lean είναι η εξάλειψη της σπατάλης σε όλους τους τομείς και φυσικά και στην ανθρώπινη προσπάθεια. Υπάρχουν πολλές εταιρείες και managers που θεωρούν lean τη βελτιστοποίηση των διαδικασιών με ταυτόχρονη απόλυση υπαλλήλων. Λόγω αυτής της αντιμετώπισης, το lean χαρακτηρίζεται από πολλούς ως “**mean**”. Φυσικά ο στόχος του lean δεν έχει καμία σχέση με απολύσεις, αν και θα μπορούσαν να υπάρχουν. Οι εταιρείες που χρησιμοποιούν τη lean φιλοσοφία ως όχημα απολύσεων, δεν έχουν όραμα ανάπτυξης ώστε να χρησιμοποιήσουν τους αποδεσμευμένους ανθρώπινους πόρους σε κάποιον άλλον τομέα, απλά προσπαθούν να επιβιώσουν. Η στασιμότητα όμως δεν είναι επιτυχημένη στρατηγική ούτε για την εταιρεία ούτε για τους υπαλλήλους της.

Το Lean πρεσβεύει την εξέλιξη. Οι άνθρωποι θα πρέπει να εκπαιδεύονται και να αποκτούν προσόντα, ώστε να καλύψουν πραγματικές ανάγκες και όχι να διατηρούν «σπάταλες» θέσεις εργασίας απλά για να απασχολούνται. Αυτή είναι μια προοδευτική κουλτούρα που θα έπρεπε να καλλιεργεί ο καθένας ατομικά, οι εταιρείες και οι οργανισμοί, ακόμα και τα κράτη.

### **2.5.2 Δυσκολία εφαρμογής.**

Επειδή η φιλοσοφία του Lean δεν είναι εύπεπτη ούτε εύκολα υλοποιήσιμη, πολλές εταιρείες έχουν πέσει στην παγίδα του να αντιγράψουν πρακτικές από άλλες χωρίς επιτυχία. Ενώ στα αρχικά στάδια εφαρμογής του lean υπάρχει ενθουσιασμός και θετικό πνεύμα, το όλο εγχείρημα καταλήγει σε αποτυχία με μερική εφαρμογή ή και ακύρωσή του. Οι βασικότεροι λόγοι μιας πιθανής αποτυχίας είναι :

- Μη ύπαρξη στρατηγικής από την ανώτερη διοίκηση.
- Προσπάθεια αντιγραφής πρακτικών από άλλους οργανισμούς.
- Η λάθος αντίληψη ότι το Lean είναι ένα set εργαλείων και μόνο.
- Η μη συμμετοχή των εργαζομένων στις διαδικασίες βελτίωσης
- Το να μη δίνεται η δυνατότητα στους εργαζόμενους να εκφράσουν απόψεις, προβληματισμούς και προτάσεις
- Η ελλιπής εκπαίδευση των εργαζομένων

Θα πρέπει να γίνει κατανοητό σε όποια εταιρεία αποφασίσει να εφαρμόσει τη Lean Φιλοσοφία ότι δεν είναι άλλη μια μέθοδος βελτίωσης εσόδων-εξόδων, αλλά στην πραγματικότητα είναι στάση ζωής.



## ΚΕΦΑΛΑΙΟ 3. LEAN SERVICES / LEAN IT

### 3.1 Services και Lean.

Οι αρχές του Lean, μετά από τη βιομηχανία, έχουν βρει εφαρμογή και στις υπηρεσίες, στα logistics, στις τράπεζες, στις ασφαλιστικές εταιρείες, στα νοσοκομεία, στην εστίαση, στις αερομεταφορές, στο IT κλπ.

Θεωρητικά εφόσον η κεντρική ιδέα είναι η εξάλειψη της σπατάλης, αρκεί να βρεθούν τα περιττά βήματα στην αλυσίδα αξίας και να καταργηθούν/αντικατασταθούν/βελτιωθούν.

Παρά την κοινή κεντρική ιδέα, υπάρχουν βασικές διαφορές ανάμεσα στη βιομηχανία και τις υπηρεσίες, που οδηγούν σε διαφορετική προσέγγιση την εφαρμογή του lean. Οι σημαντικότερες είναι οι παρακάτω:

- η σπατάλη και η δημιουργία αξίας είναι αρκετά ορατές (visible) στην παραγωγή σε αντίθεση με τις υπηρεσίες. Τα σημεία συμφόρησης (bottlenecks) και η αναμονή/αδράνεια είναι πιο αναγνωρίσιμες και χειροπιαστές στην παραγωγή. Π.χ. σωροί από αντικείμενα που περιμένουν κάποιον να τα χρησιμοποιήσει ή εργάτες που περιμένουν για να προχωρήσουν στο επόμενο βήμα. Στις υπηρεσίες ούτε η αξία που δημιουργείται είναι ορατή, ούτε οι χρόνοι αναμονής/αδράνειας π.χ. όταν ο υπάλληλος που αδρανεί βρίσκεται μπροστά σε ένα PC. Συνήθως η συμφόρηση και η πίεση γίνονται αντιληπτά μόνο όταν κάποιος υπάλληλος παραπονεθεί για τον υπερβολικό φόρτο εργασίας. Ειδικά για αυτούς τους λόγους είναι απαραίτητη η αποτύπωση και η μελέτη του value stream ώστε να αναγνωριστούν οι σπατάλες και τα σημεία συμφόρησης.
- Οι διαδικασίες της παραγωγής δεν περιλαμβάνουν ούτε απαιτούν την άμεση επικοινωνία με τον πελάτη. Η αγορά ενός τηλεφώνου π.χ. δεν προϋποθέτει την επαφή του πελάτη με τη διαδικασία της παραγωγής του τηλεφώνου. Αντίθετα ένα τραπεζικό δάνειο ή η αλλαγή ενός SW προγράμματος, απαιτεί πολλές και συχνές άμεσες επικοινωνίες του εργαζόμενου με τον πελάτη και οι δραστηριότητες του εργαζόμενου έχουν άμεσο αντίκτυπο στον πελάτη. Ουσιαστικά στον τομέα των υπηρεσιών θα μπορούσε να πει κάποιος πως η διαδικασία είναι το προϊόν.
- το ανθρώπινο δυναμικό σε έναν οργανισμό που πουλάει υπηρεσίες παρουσιάζει μεγάλη διαφοροποίηση τόσο σε προσόντα όσο και σε εξειδίκευση, πράγμα που δυσκολεύει προσπάθειες τυποποίησης διαδικασιών.

- Η παραγωγή βασίζεται σε μηχανήματα και Α ύλες κι εκεί εντοπίζονται τα μεγαλύτερα κόστη, σε αντίθεση με τις υπηρεσίες όπου είναι το ανθρώπινο δυναμικό παράγοντας κόστους.

Επειδή το μεγαλύτερο βάρος στις υπηρεσίες κατέχει ο ανθρώπινος παράγοντας, πολλοί μελετητές εστιάζουν στην ενίσχυση του εργατικού δυναμικού μέσω του lean, τόσο στην εκπαίδευση όσο και στην αύξηση της αυτονομίας του (empowerment). (Bowen, D. E., & Youngdahl, W. E., 1998).

Όσον αφορά στην εφαρμογή των αρχών του lean στις υπηρεσίες, για τους Womack & Jones (2005) είναι απαραίτητη η εφαρμογή αρχών όπως : Πλήρης επίλυση του προβλήματος του πελάτη, διασφαλίζοντας ότι όλες οι υπηρεσίες που αφορούν το πρόβλημα του πελάτη λειτουργούν ορθά και συντονισμένα ώστε να μη σπαταλάνε το χρόνο του πελάτη, του παρέχουν ακριβώς αυτό που θέλει εκεί που το θέλει και την ώρα που το θέλει. Θα αναφερθούμε σε αυτές τις αρχές εκτενέστερα στο κεφάλαιο 5. Ο Maister (1985) πρότεινε δύο «νόμους υπηρεσιών» (Service Laws): Ο πρώτος συγκρίνει τις προσδοκίες του πελάτη με την αντίληψή του για την παρεχόμενη υπηρεσία. Εάν η αντίληψή του είναι καλύτερη από την προσδοκία του, τότε ο πελάτης είναι ευχαριστημένος.

Ο δεύτερος νόμος υποστηρίζει ότι η πρώτη εντύπωση του πελάτη μπορεί να επηρεάσει τη συνολική εμπειρία της παροχής υπηρεσίας.

Όσον αφορά στα εργαλεία του lean η κάθε εταιρεία χρησιμοποιεί εκείνα που την εξυπηρετούν. Σε μεγάλο βαθμό όμως χρησιμοποιούνται το Value Stream Mapping, η εξισορρόπηση φόρτου εργασίας (heijunka) και το 5S. (Higor Leite, 2015)

Στις υπηρεσίες πολλές φορές σχεδιάζεται η συμμετοχή του πελάτη στην υλοποίηση των lean αρχών. Χαρακτηριστικό παράδειγμα οι αεροπορικές εταιρείες χαμηλών ναύλων. Για να μειώσουν όσο γίνεται το χρόνο παραμονής του αεροπλάνου στο έδαφος, ζητάνε από τους πελάτες να μαζέψουν οι ίδιοι τα περιοδικά και τα σκουπίδια τους. Έτσι μειώνουν τόσο το κόστος παραμονής στο έδαφος, όσο και τα έξοδα καθαριότητας. Άλλο ένα παράδειγμα lean logistics στις αερομεταφορές είναι ο προγραμματισμός του χρόνου επιβίβασης. Οι επιβάτες καλούνται σε διαφορετική ώρα για την επιβίβαση ανάλογα με τη θέση τους στο αεροπλάνο.

Στο χώρο της υγείας, το lean έχει εφαρμοστεί κυρίως σε νοσοκομεία με στόχο τη μείωση της αναμονής αλλά και την αύξηση της ποιότητας των παρεχόμενων υπηρεσιών όπως την αντιλαμβάνεται ο πελάτης. Ένα τέτοιο παράδειγμα είναι το LifeCare Hospital, το οποίο

Athens MBA, Εφαρμογή της Lean Φιλοσοφίας σε εταιρικό Τμήμα Ανάπτυξης Λογισμικού

πέτυχε τη μείωση των λοιμώξεων από μεταγίσεις κατά 70%, μετά την εφαρμογή του lean. (Higor Leite, 2015)

Αναφέρουμε μερικές γνωστές εταιρείες που έχουν εφαρμόσει Lean με μεγάλη επιτυχία :

Η Wal-Mart, εξάλειψε τα αποθέματα στα καταστήματά της και στις αποθήκες.

Η Zara, που προσφέρει τεράστια ποικιλία ρουχισμού, χωρίς να κρατάει μεγάλα αποθέματα και με εξαιρετική ανταπόκριση στις απαιτήσεις της αγοράς του ενδύματος.

Η Southwest Airlines, εξάλειψε τα κόστη υπηρεσιών, με παράλληλη βελτίωση της εμπειρίας των πελατών.

Η Jefferson Pilot Financial Company, βελτίωσε τις λειτουργίες της και αύξησε τα κέρδη. Μείωσε κατά 70% το χρόνο επεξεργασίας απαίτησης και 26% το εργατικό κόστος

Στη συνέχεια θα αναφερθούμε εκτενέστερα στο IT (Information Technology).

### **3.2 Παρούσα κατάσταση στο IT.**

Στη σημερινή εποχή το IT παίζει σημαντικό ρόλο στην επιτυχία μιας επιχείρησης. Αυτό δεν αποτρέπει δυστυχώς τις εξαιρετικά υψηλές δαπάνες σε ανεπαρκή IT projects. Οι συνέπειες τέτοιων projects είναι ανυπολόγιστες διότι αφήνουν πολλά προβλήματα τα οποία για να λυθούν απαιτούν(ξανά) χρήμα, κόπο και πόρους (ανθρώπους αλλά και μηχανήματα).

Τα κεφάλαια της παγκόσμιας οικονομίας που δαπανώνται στο IT είναι σημαντικά. Το 2008 π.χ. ο χρηματοπιστωτικός τομέας ξόδεψε 6,9% των κερδών τους στο IT, ενώ ο τομέας της υγείας το 4,7%(M.Orzen,2010). Το αρνητικό εδώ είναι ότι μεγάλο ποσοστό αυτών των χρημάτων (20%) δεν προσθέτει αξία στον πελάτη. Το χειρότερο όλων όμως είναι οι συνέπειες κακών και αναποτελεσματικών IT συστημάτων στην παραγωγικότητα και στη γενικότερη απόδοση του οργανισμού, και κατ' επέκταση στην παγκόσμια οικονομία.

Το Standish Group συγκεντρώνει σε ετήσια βάση από το 1994 πληροφορίες για την επιτυχία των IT projects, με απώτερο στόχο τη βελτίωση του χώρου. Το report αυτό

ονομάζεται CHAOS Report (σύμφωνα με μια εξήγηση είναι το ακρωνύμιο του Comprehensive Human Appraisal for Originating Software). Όπως φαίνεται ένα μεγάλο ποσοστό, περίπου 20% των projects αποτυγχάνουν, ακυρώνονται ή παραδίδονται αλλά ποτέ δε χρησιμοποιούνται, περίπου 50% παραδίδονται καθυστερημένα ή με υπέρβαση του προϋπολογισμού(budget) ή χωρίς όλη τη λειτουργικότητα που είχαν υποσχεθεί. Μόνο 30% καταφέρνουν να είναι επιτυχή, με έγκαιρη παράδοση και πλήρως λειτουργικά.

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Πίνακας 4. CHAOS Report

Η βασική αιτία αυτής της τεράστιας αποτυχίας των IT projects είναι ότι το Business και το IT δεν έχουν απόλυτη και ξεκάθαρη μεταξύ τους επικοινωνία. Παρακάτω βλέπουμε την άποψη του Business για τη συνεργασία με το IT, αλλά και το αντίστροφο (S.Bell, M.Orzen, 2010) :

### 3.3 Οπτική Business

- Πολυπλοκότητα:

Τα συστήματα εφαρμογών που παρέχει το IT είναι συχνά δύσχρηστα, κοστοβόρα και δυσκίνητα (δεν μπορούν να αλλάξουν/εξελιχτούν εύκολα).

- Ταχύτητα:

Το IT θεωρείται βραδυκίνητο ειδικά για τις έκτακτες και επείγουσες αλλαγές/απαιτήσεις.

- Αποπροσανατολισμός:

Το IT εστιάζει περισσότερο σε τεχνικά θέματα παρά στην πραγματική επίλυση των προβλημάτων του Business.

- Διαφορετική γλώσσα:

Το IT μιλάει γλώσσα που δεν καταλαβαίνει το business, επίσης όμως και το business δεν γίνεται κατανοητό από τους IT επαγγελματίες.

- Υπερπαραγωγή πληροφορίας:

Στην προσπάθειά του να κατανοήσει το Business, το IT παράγει υπερβάλλουσα πληροφορία πολλά mails, κείμενα documentation κλπ.

- Αποτυχία Project:

Τα IT projects κοστίζουν, είναι χρονοβόρα, αργούν και προκαλούν αναστάτωση, ενώ πολλές φορές αποτυγχάνουν στους στόχους τους.

- Κατακερματισμός συστημάτων:

Συχνά υπάρχουν πολλά διαφορετικά συστήματα που συνδέονται μεταξύ τους για την ολοκλήρωση μιας διαδικασίας.

- Κακή ποιότητα δεδομένων:

Πολλές φορές τα δεδομένα και οι πληροφορίες που παρέχονται είναι ανακριβή, και αναξιόπιστα.

- Ανεπαρκής υποστήριξη αποφάσεων:

Οι χρήστες νιώθουν σύγχυση διότι έχουν πολλά δεδομένα αλλά λίγη πραγματικά χρήσιμη πληροφορία και μάλιστα τη στιγμή που τη χρειάζονται και με το format που τους διευκολύνει, ώστε να βοηθηθούν στη λήψη των αποφάσεων.

- Αναρχία συστημάτων:

Πολλοί χρήστες διατηρούν δικά τους αρχεία με πληροφορίες, στην προσπάθειά τους να έχουν την πληροφόρηση που θέλουν(workaround), κι έτσι συμβάλλουν στον περαιτέρω κατακερματισμό των συστημάτων.

- Εστίαση στο κόστος:

Συχνά το IT θεωρείται ως back office Κέντρο Κόστους, και όχι ως τμήμα που ασχολείται με την καινοτομία, τον πειραματισμό και την παραγωγή αξίας.

- Δυσκολία υπολογισμού της απόδοσης επένδυσης(ROI):

Το business δεν μπορεί να υπολογίσει επακριβώς την απόδοση των επενδύσεων σε πληροφοριακά συστήματα (hardware, software) και να αξιολογήσει την ποιότητα και την αποτελεσματικότητα της απόδοσης του IT. Αυτή η ασάφεια οδηγεί σε λάθος αντίληψη της πραγματικής αξίας του IT.

### 3.4 Οπτική IT.

Στη συνέχεια θα δούμε τα πράγματα από τη μεριά του IT, το οποίο τις περισσότερες φορές είναι υπερφορτωμένο και αντιμετωπίζει συνεχείς αλλαγές προτεραιοτήτων και απαιτήσεων από τους χρήστες. Παράλληλα πρέπει να συμβαδίζει με την εξέλιξη της τεχνολογίας φροντίζοντας για νέες εκδόσεις και αναβαθμίσεις εφαρμογών (releases, upgrades). Το αποτέλεσμα τις περισσότερες φορές είναι να λειτουργεί στα προβλήματα αντιδραστικά (reactively), και να δημιουργούνται συστήματα ασταθή και περίπλοκα. Οι εκκρεμότητες του IT, ειδικά στο Software Development, είναι μεγάλες. Πολλές εταιρείες αναφέρουν έως και πέντε χρόνια καθυστέρηση στην υλοποίηση αιτημάτων τους. Και ταυτόχρονα τα κόστη πρέπει να μένουν όσο πιο χαμηλά γίνεται. Όλη αυτή η κατάσταση προκαλεί στους εργαζόμενους στο IT μεγάλη πίεση και άγχος. Παρακάτω αναφέρονται τα σημαντικότερα θέματα που αντιμετωπίζει το IT και ιδιαίτερα το SW Development :

- Πυροσβεστική επίλυση προβλημάτων:

Συχνά οι έκτακτες δουλειές που προκύπτουν και πρέπει να γίνουν είναι πιο πολλές από τις προγραμματισμένες. Αυτό οδηγεί σε εκνευρισμό, κούραση και τελικά είναι μη βιώσιμη πρακτική.

- Ασαφείς προδιαγραφές απαιτήσεων:

Οι χρήστες δεν μπορούν να διατυπώσουν σαφώς και με ευκρίνεια την απαίτησή τους και πολύ συχνά ζητάνε περισσότερα πράγματα από αυτά που πραγματικά έχουν ανάγκη.

- Αντικρουόμενες προτεραιότητες:

Οι business owners δεν μπορούν να συμφωνήσουν στις προτεραιότητες που ορίζουν με αποτέλεσμα το IT να βρίσκεται στη «μέση», προσπαθώντας να ισορροπήσει τις καταστάσεις.

- Καθυστερημένη (Ελλιπής) συμμετοχή στα projects :

Είναι συχνό φαινόμενο να λαμβάνονται οι αποφάσεις για σημαντικά project που πρέπει να υλοποιήσει το IT, χωρίς να έχει συμμετάσχει στις αρχικές συζητήσεις.

- Εξάντληση των ανθρώπινων πόρων:

Λόγω των απρόβλεπτων έκτακτων απαιτήσεων που λαμβάνει το IT, που συνοδεύονται από ασαφείς και μεταβλητές προτεραιότητες, οι εργαζόμενοι στο IT δεν προλαβαίνουν να ολοκληρώσουν τη μια δουλειά και πιάνουν την άλλη. Αυτό προφανώς αυξάνει τους χρόνους υλοποίησης, αυξάνοντας ταυτόχρονα το κόστος, μειώνει την παραγωγικότητα, προκαλεί λάθη και ποιοτικά προβλήματα, ενώ παράλληλα κουράζει και απογοητεύει τους εργαζόμενους .

- Εκτεταμένοι αυτοματισμοί:

Αντί της απλοποίησης χρονοβόρων ή «σπάταλων» διαδικασιών, δηλ. που περιέχουν άχρηστα βήματα (muda), αυτές αυτοματοποιούνται δίνοντας την ψευδαίσθηση βελτίωσης, ενώ στην πραγματικότητα αυξάνουν την πολυπλοκότητα.

- Κακή ποιότητα δεδομένων:

Οδηγεί σε λάθη, διορθώσεις προγραμμάτων και συχνά προκαλείται λόγω έλλειψης εκπαίδευσης των τελικών χρηστών, έλλειψης τεκμηρίωσης (documentation) και κακού σχεδιασμού των διαδικασιών και των ελέγχων.

- Προγραμματισμός των πόρων:

Ο ορθολογικός προγραμματισμός των πόρων ειδικά των ανθρώπινων είναι μια συνεχής πρόκληση. Οι προγραμματιστές και οι σύμβουλοι μοιράζουν το χρόνο τους σε πολλά έργα και λειτουργίες, που έχουν αντικρουόμενες προτεραιότητες, προκαλώντας «ουρές» και καθυστερήσεις.

- Περιορισμένος προϋπολογισμός:

Το IT αντιμετωπίζεται ως Κέντρο Κόστους που πρέπει να μειώσει τα έξοδά του, ενώ θα έπρεπε να αντιμετωπίζεται ως τμήμα που δημιουργεί αξία, καινοτομία και βελτιώνει τις επιχειρησιακές διαδικασίες.

- Outsourcing (εξωτερικοί συνεργάτες):

Το business πιστεύει ότι το outsourcing διοικητικών λειτουργιών και των IT υπηρεσιών θα μειώσει το κόστος. Η μείωση όμως του κόστους δε σημαίνει πάντα θετικά

αποτελέσματα. Είναι πολύ σημαντικό για μια εταιρεία να δημιουργεί ανταγωνιστικό πλεονέκτημα και οι Άνθρωποι της θα μπορούσαν να είναι ακριβώς αυτό. Αυτή τη λογική άλλωστε ακολουθούσε και η Toyota.

Από την παραπάνω καταγραφή της τωρινής κατάστασης ένα πράγμα είναι απόλυτα ξεκάθαρο: στο χώρο του IT και ειδικά στο Software Development, υπάρχει ανάγκη και πρόσφορο έδαφος για ριζική αναθεώρηση της υφιστάμενης κατάστασης μέσω της Lean φιλοσοφίας.



### 3.5 Πώς το Lean IT δημιουργεί αξία.

Ένας Lean οργανισμός θα πρέπει πρώτα από όλα να απλοποιεί τις διαδικασίες και στη συνέχεια αν γίνεται, να αυτοματοποιεί επαναλαμβανόμενες εργασίες ρουτίνας. Η βελτίωση των διαδικασιών ελευθερώνει ανθρώπινο δυναμικό που θα μπορεί να πειραματιστεί για να βρει καλύτερες και καινοτόμες λύσεις σε βασικότερα προβλήματα.

Το IT και το Business πρέπει να συνεργαστούν ώστε να αντιμετωπίσουν τα μεγάλα θέματα της ευθυγράμμισης της εταιρικής στρατηγικής με την καθημερινότητα παντού μέσα στον οργανισμό. Αυτή η ευθυγράμμιση γίνεται σε τρεις διαστάσεις(S.Bell, M.Orzen, 2010) :

1. Κάθετα από πάνω προς τα κάτω. Από την Κορυφή της Διοίκησης προς κάθε Τμήμα ώστε ο κάθε εργαζόμενος να αντιλαμβάνεται πώς η δική του καθημερινή συμμετοχή υποστηρίζει το κοινό όραμα, την εταιρική στρατηγική, τους σκοπούς και τους στόχους του οργανισμού.
2. Οριζόντια, δίνοντας έμφαση στην απρόσκοπτη «οριζόντια» δημιουργία αξίας για τον πελάτη (flow), διασπώντας τα λειτουργικά σιλό (τμήματα), αντί της συνηθισμένης προσπάθειας βελτιστοποίησης ανά λειτουργικό σιλό (τμήμα).
3. Κάθετα και οριζόντια ευθυγραμμίζοντας τα συστήματα πληροφοριών, ηλεκτρονικά ή μη, διασφαλίζοντας ότι το IT επιτρέπει την υλοποίηση της εταιρικής στρατηγικής και προσθέτει αξία στις διαδικασίες του business, στα projects και στη διοίκηση. Ο συγχρονισμός της κάθετης και οριζόντιας ευθυγράμμισης είναι το σημείο ισορροπίας για τη βελτίωση των διαδικασιών στις οποίες συμμετέχει το IT (είτε ως πάροχος υπηρεσίας, είτε στις εσωτερικές του).

Από τα παραπάνω γίνεται αντιληπτό ότι η εφαρμογή του Lean στο IT λειτουργεί σε δύο αλληλένδετα επίπεδα:

- Στο **εξωτερικό**, που αφορά τη συνεργασία με το business και τη συνεχή βελτίωση των διαδικασιών του business. Ο στόχος είναι η αύξηση της αξίας προς τον πελάτη (business) με την παροχή ολοκληρωμένης και ορθής πληροφορίας, με ταυτόχρονη μείωση του κόστους. Έτσι βοηθάει στη βελτίωση της παραγωγικότητας, της ανταγωνιστικότητας, της ποιότητας και κατά συνέπεια της κερδοφορίας του οργανισμού.
- Στο **εσωτερικό**, που αφορά την εφαρμογή των Lean αρχών στο ίδιο το IT, στις εσωτερικές του διαδικασίες, στο software development και στα Projects που αναλαμβάνει ή στα οποία συμμετέχει το IT.

Εν κατακλείδι μπορούμε να πούμε ότι το Lean IT χρησιμοποιεί τις αρχές και τα εργαλεία του Lean, ώστε να ευθυγραμμίσει και να συγχρονίσει το IT με το Business, με απώτερο σκοπό την παροχή ποιοτικής πληροφόρησης, αποτελεσματικών Πληροφοριακών Συστημάτων και τη συνεχή βελτίωση των διαδικασιών του οργανισμού.

### 3.6 Lean και Green IT.

Παρόλο που δεν είναι στους άμεσους στόχους αυτής της εργασίας, θεωρώ ότι είναι σημαντικό να γίνει αναφορά στο ρόλο που μπορεί να παίξει το IT στη μείωση του περιβαλλοντικού αποτυπώματος του οργανισμού και στην καλύτερη ενεργειακή διαχείριση. Στην εποχή μας που μαστίζεται από την κλιματική αλλαγή, το Lean IT και το Green IT είναι συμπληρωματικά. Στην πραγματικότητα μετά από μια επιτυχημένη Lean προσπάθεια, είναι πολύ πιθανό να υπάρχουν περιβαλλοντικά οφέλη χωρίς να έχουν προδιαγραφεί ή συζητηθεί εκ των προτέρων. Ένα τέτοιο παράδειγμα στις επιχειρήσεις που κάνουν μεταφορές, είναι η βελτίωση της διαδικασίας δρομολόγησης. Ενώ ο στόχος είναι η μείωση των εξόδων από καύσιμα, παράλληλα μειώνεται η εκπομπή CO<sub>2</sub>.

Για να επιτευχθεί το Green IT, καταρχήν θα πρέπει το ίδιο το IT να βρει μεθόδους και διαδικασίες ώστε να μειώσει το δικό του ενεργειακό αποτύπωμα ελέγχοντας και βελτιώνοντας :

1. Την ενεργειακή απόδοση των Data Centers
2. Την ενεργειακή απόδοση του τελικού χρήστη
3. Την κατανάλωση μελάνης και εκτυπωτικού χαρτιού
4. Την ορθή χρήση/αποκομιδή των ηλεκτρονικών αποβλήτων

Επιπλέον μείωση περιβαλλοντικών συνεπειών μπορούν να επιτευχθούν με τη βελτίωση των διαδικασιών του Business, όπως είδαμε στο παράδειγμα της δρομολόγησης. Ανάλογες βελτιώσεις μπορούν να υλοποιηθούν σε διαδικασίες που δημιουργούν π.χ. εκτυπώσεις ή archiving αρχείων, γλιτώνοντας πολύτιμο χαρτί, αποθηκευτικό χώρο φυσικό/ηλεκτρονικό κλπ.

Το IT έχει τη δυνατότητα να βοηθήσει την εταιρεία μέσω reporting, ώστε να έχει εικόνα του πόσο επηρεάζουν οι διαδικασίες το περιβαλλοντικό αποτύπωμά της και να λάβει τις κατάλληλες αποφάσεις. Στο παράδειγμα της δρομολόγησης π.χ. όταν γνωρίζουμε την κατανάλωση καυσίμου, γνωρίζουμε αυτομάτως και την εκπομπή CO<sub>2</sub>.

## ΚΕΦΑΛΑΙΟ 4 . LEAN ΑΡΧΕΣ ΣΤΟ SW DEVELOPMENT

### 4.1 To SW Development ως προϊόν.

Το SW Development είναι μια διαδικασία παραγωγής γνώσης και ανάπτυξης προϊόντος. Στην πραγματικότητα ο χρήστης του κώδικα που δημιουργείται από τον προγραμματιστή, αγοράζει το προϊόν και τις ιδιότητές του, π.χ. το ηλεκτρονικό παιχνίδι, τον επεξεργαστή κειμένου κλπ. Σε μια εταιρεία που έχει εσωτερικό τμήμα ανάπτυξης κώδικα, το προϊόν που δημιουργείται και «αγοράζει» ο χρήστης καλύπτει την ανάγκη μιας εταιρικής διαδικασίας.

Οι αρχές του Lean Manufacturing βρίσκουν εφαρμογή και στο SW Development, με την κατάλληλη βέβαια προσαρμογή διότι το development έχει διαφορετική υπόσταση. Στο manufacturing γνωρίζουμε ακριβώς τι πρέπει να δημιουργηθεί και με ποιο τρόπο. Η δημιουργία όμως ενός προϊόντος γνώσης ή η ανάπτυξη ενός νέου προϊόντος, έχει να κάνει περισσότερο με τον πειραματισμό, ξεκινώντας από την ιδέα και καταλήγοντας στο τελικό προϊόν, μέσα από μια πορεία διερεύνησης, προσαρμογών και αλλαγών. Αυτήν την εμπειρική προσέγγιση ακολουθεί και η Toyota στο σύστημα ανάπτυξης νέων προϊόντων, ξεκινώντας από την ιδέα και τις βασικές προδιαγραφές/απαιτήσεις για ένα νέο αυτοκίνητο, εμπλουτίζοντας στην πορεία το δημιουργούμενο προϊόν με τα κατάλληλα χαρακτηριστικά, έως ότου λάβει την τελική του μορφή. Μάλιστα σύμφωνα με τον Kosaku Yamada, chief engineer του Lexus ES 300 : «Η πραγματική διαφορά ανάμεσα στην Toyota και τους άλλους κατασκευαστές αυτοκινήτων δεν είναι το Toyota Production System. Είναι το **Toyota Product Development System**». (Gary S.Vasilash, 2001).

Για να γίνει πιο κατανοητή η διαφοροποίηση ανάμεσα στην ανάπτυξη και στη βιομηχανία, θα χρησιμοποιήσουμε το παράδειγμα της δημιουργίας(ανάπτυξης) μιας καινούργιας συνταγής από έναν Chef (Ballard, 2000). Η δημιουργία της συνταγής για ένα νέο πιάτο είναι μια διαδικασία γνώσης, δοκιμής και λάθους(trial and error). Δεν είναι αναμενόμενη η επιτυχία από την πρώτη δοκιμή. Για να δημιουργήσει τη συνταγή, θα πρέπει να δοκιμάσει πολλές παραλλαγές και να κατασταλάξει στην επικρατέστερη.

Από τη στιγμή που ο Chef έχει δημιουργήσει τη συνταγή, το να τη μαγειρέψει κάποιος σημαίνει ότι ακολουθεί και επαναλαμβάνει πιστά τις οδηγίες και τα βήματα της συνταγής. Αυτή η διαδικασία έχει να κάνει με τη βιομηχανία και την παραγωγή.

Στον παρακάτω πίνακα βλέπουμε τις διαφορές ανάμεσα στο Lean Manufacturing και στο Lean Development:

<b>Lean Manufacturing</b>	<b>Lean Development</b>
Η ποιότητα καθορίζεται βάσει προδιαγραφών	Η ποιότητα καθορίζεται από την επίτευξη του σκοπού
Η μεταβλητότητα στο αποτέλεσμα δεν είναι επιθυμητή. Όλα τα προϊόντα που παράγονται πρέπει να είναι ίδια και με τα ίδια χαρακτηριστικά (του ενός μοναδικού Πρωτοτύπου)	Η μεταβλητότητα είναι θεμιτή, διότι θα καταλήξει στο ένα μοναδικό Πρωτότυπο, (μέσα από σχεδιασμό ημιτελών προϊόντων)

Πίνακας 5. Διαφορές Lean Manufacturing / Lean Development

Στη συνέχεια θα αναφερθούμε στις βασικές Lean αρχές που πρέπει να διέπουν το SW Development και οι οποίες μπορούν να εφαρμοστούν και γενικότερα στην ανάπτυξη νέων προϊόντων. (Mary and Tom Poppendieck, 2007)

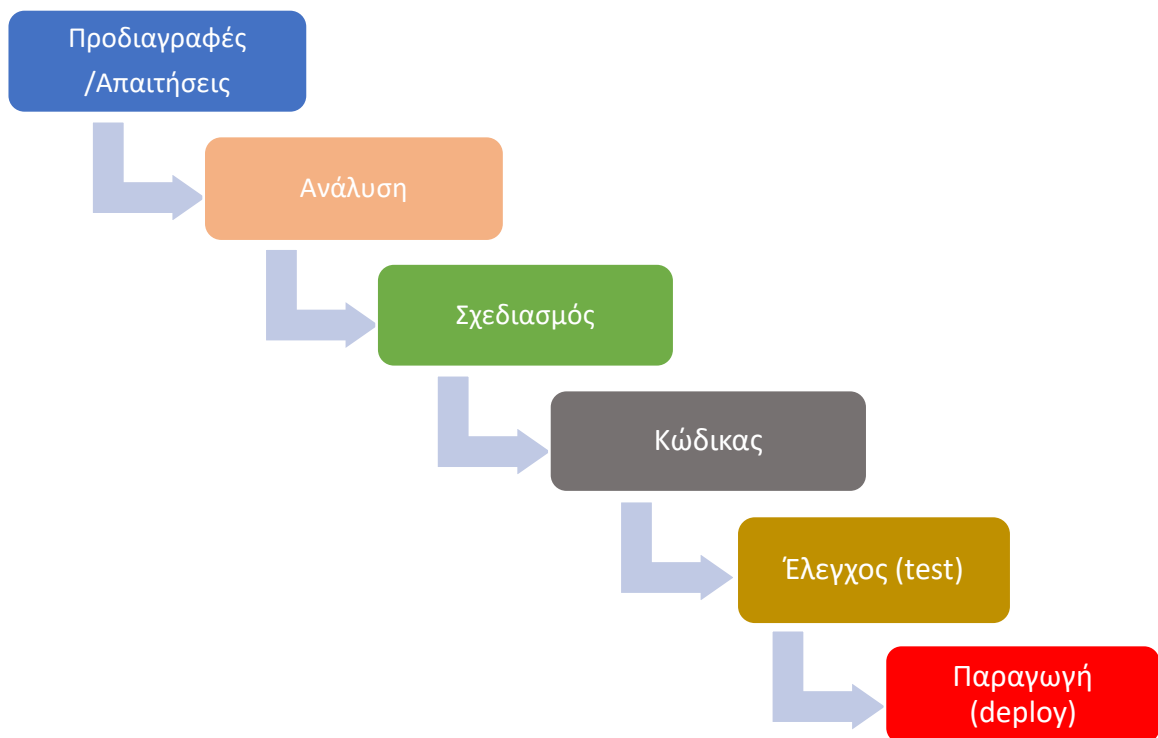
#### **4.2 Εξάλειψη της σπατάλης.**

Ο στόχος είναι η αναγνώριση και εξάλειψη των διαδικασιών ή των σημείων στη διαδικασία που δεν έχουν αξία για τον πελάτη. Αυτό σημαίνει ότι πρέπει να είναι ξεκάθαρη αυτή η έννοια της αξίας. Στη βιομηχανία του SW η αξία που αναγνωρίζει ο πελάτης αλλάζει συχνά διότι τις περισσότερες φορές ο ίδιος ο πελάτης δε γνωρίζει τι ακριβώς θέλει. Επίσης είναι σίγουρο ότι η άποψη του πελάτη για την αξία του SW που παραλαμβάνει θα μεταβληθεί όταν το χρησιμοποιήσει παραγωγικά στην καθημερινότητά του (προς το καλύτερο ή προς το χειρότερο).

Όσον αφορά στην αναγνώριση της σπατάλης, όπως και στη βιομηχανία έτσι και στο SW Development ένα είδος σπατάλης είναι τα αποθέματα. Στο SW ως απόθεμα εννοείται η ημιτελής δουλειά. Αναφερόμαστε σε αλλαγές που ξεκινάνε από κάποιες απαιτήσεις, φτάνουν σε ένα σημείο ελέγχου από τους χρήστες και εκεί «κολλάνε» (User Action), λόγω πιθανής υπερφόρτωσης των χρηστών με καθημερινές εργασίες ή αλλαγής των προτεραιοτήτων τους.

Άλλη βασική σπατάλη είναι οι αλλαγές των προδιαγραφών λόγω κακού χρονισμού και καθυστερήσεων: όταν ο κώδικας γράφεται πολύ μετά από τις αρχικές προδιαγραφές, ή όταν το test γίνεται πολύ μετά από τη συγγραφή του κώδικα, είναι πολύ λογικό να έχουν υπάρξει μεταβολές στις πραγματικές απαιτήσεις του χρήστη. Πράγμα που σημαίνει (ξανά) γράψιμο κώδικα, ελέγχους, κόπο και κόστος.

Αυτή η προσέγγιση του waterfall (καταρράκτη) φαίνεται στο σχήμα 9, όπου ο χρόνος ανάμεσα στην αρχική απαίτηση του πελάτη/χρήστη και στην τελική παράδοση του SW είναι τεράστια (μήνες ή και χρόνια). Η προσέγγιση αυτή πηγάζει από την αντίληψη της μαζικής παραγωγής όπου μεγάλες παρτίδες αλλαγών συγκεντρώνονται και μπαίνουν σε σχεδιασμένες και ελεγχόμενες εκδόσεις (releases).



Σχήμα 9. Waterfall approach

Τα αποτελέσματα όμως της μακράς διάρκειας των projects, είναι ολέθρια: χαμένος χρόνος λόγω μετακινήσεων των προγραμματιστών/πόρων σε διάφορα έργα, αλλαγές προδιαγραφών, καθυστερήσεις και ελλιπή test, μεταφορά λαθών στην παραγωγή, εντοπισμός των λαθών στην παραγωγή, διόρθωση των λαθών εν μέσω παραγωγικής διαδικασίας. Στις συνέπειες συγκαταλέγονται, εκτός από τον κόπο και το χαμένο χρόνο και η τρομερή αναστάτωση που προκαλείται σε όλους τους εμπλεκόμενους. Το χειρότερο

όμως είναι ότι πολλές φορές τα λάθη έχουν αντίκτυπο σε εφαρμογές που δεν είχαν άμεση συμμετοχή στις αλλαγές, προκαλώντας ακόμα περισσότερα προβλήματα.

Η σημαντικότερη όμως σπατάλη είναι η μεγάλη λειτουργικότητα (function/features) που χτίζεται στα προγράμματα, και που **ποτέ** δεν θα χρησιμοποιηθεί. Υπολογίζεται ότι μόνο γύρω στο 20% της λειτουργικότητας των custom προγραμμάτων χρησιμοποιείται από τους χρήστες. Αυτή η σπατάλη προκαλεί μεγάλη πολυπλοκότητα στον κώδικα και τεράστιο κόστος συντήρησής του.

Ποιά είναι η αιτία για αυτήν την τεράστια παραγωγή «άχρηστου» στην ουσία κώδικα; Έχει να κάνει με τη λάθος(παραδοσιακή) προσέγγιση των SW projects. Η συνηθισμένη τακτική είναι ως πρώτο βήμα να καθορίζονται οι απαιτήσεις από τον πελάτη/χρήστη και στη συνέχεια να υπογράφονται με σκοπό να αποτελέσουν το score του έργου, από το οποίο δε θα πρέπει να αποκλίνει ο χρήστης. Για το λόγο αυτό οι χρήστες προσπαθούν να σκεφτούν όποια λειτουργία/λειτουργικότητα θεωρούν πιθανά χρήσιμη ώστε να την εντάξουν στο score του έργου και να μη βρεθούν στη δυσχερή θέση να ζητήσουν αλλαγή/ενίσχυση εκτός score.

Αυτή η «υπερπαραγωγή» (overproduction) είναι η χειρότερη σπατάλη. Κάθε γραμμή παραπανίσιου και άχρηστου κώδικα αυξάνει την πολυπλοκότητα, απαιτεί test, documentation και φυσικά υποστήριξη.

Η υπερπαραγωγή προκύπτει επίσης από άγνοια ή ελλιπή μεταφορά γνώσης των διαδικασιών ή και από κακό management. Είναι πολύ πιθανή η ύπαρξη προγραμμάτων που κάνουν την ίδια ή παραπλήσια δουλειά, αλλά λόγω άγνοιάς τους ζητείται η δημιουργία νέων.

Αυτό που πρέπει να γίνει είναι να χτιστεί διαδικασία με την οποία θα αναπτύσσεται εκείνο το 20% του κώδικα που θα προσδώσει το 80% της αξίας, και μόνο τότε να προχωράει κανείς στις επόμενες πιο σημαντικές λειτουργικότητες. Δεν πρέπει επ' ουδενί το score ενός SW έργου να περιλαμβάνει τα πάντα που μπορεί κάποιος να σκεφτεί ότι θα χρειαστούν, ειδικά όταν αυτό το score προέρχεται από χρήστες που δε γνωρίζουν απόλυτα τι θέλουν να κάνουν.

### 4.3 Εστίαση στην ποιότητα.

Το SW που παραδίδεται στους πελάτες/χρήστες ως προϊόν, θα πρέπει να έχει τέτοια ποιότητα που ο πελάτης να **αντιλαμβάνεται** την ισορροπία ανάμεσα στη λειτουργικότητα, στην ευκολία χρήσης, στην αξιοπιστία και στην οικονομία που ζήτησε. (**Perceived Integrity**).

Όσον αφορά στην ποιότητα του κώδικα, ο στόχος των προγραμματιστών και όλων όσοι ασχολούνται με το SW, πρέπει να είναι η δημιουργία ποιοτικού κώδικα εξαρχής, και όχι η εύρεση και διόρθωση λαθών εκ των υστέρων.

Σύμφωνα με τον Shigeo Shingo (1981), υπάρχουν 2 είδη ελέγχου (inspection): ο έλεγχος αφού εμφανιστεί το ελάττωμα(bug), και ο έλεγχος αποφυγής λαθών. Για την επίτευξη της πραγματικής ποιότητας πρέπει να υπάρχει το κατάλληλο περιβάλλον και οι συνθήκες που να μην επιτρέπουν τα λάθη εκ των προτέρων. Εάν αυτό δεν είναι εφικτό, θα πρέπει να επιθεωρείται το προϊόν(κώδικας) μετά από κάθε μικρή αλλαγή, ώστε να εντοπίζεται το λάθος όσο γίνεται πιο γρήγορα και να διορθώνεται άμεσα.

Τα συστήματα παρακολούθησης λαθών (defect tracking systems), σύμφωνα με τη Lean φιλοσοφία είναι σπατάλη διότι λειτουργούν καταγραφικά και δημιουργούν ουρές με δουλειά που πρέπει να γίνει, να διορθωθεί, να ελεγχτεί κλπ. Οι ουρές στο Lean είναι σημείο σπατάλης. Κανονικά δεν θα έπρεπε να υπάρχουν τέτοια συστήματα, διότι δεν θα έπρεπε να υπάρχουν τέτοια λάθη (bugs).

Ο μύθος ότι το test των χρηστών έχει σκοπό να εντοπίζει τα λάθη (bugs), πρέπει να καταρριφθεί. Στο QA σύστημα (Quality assurance), οι χρήστες πρέπει να ελέγξουν την ορθότητα και λειτουργικότητα της εφαρμογής. Δε θα πρέπει να αντιμετωπίζουν προβλήματα όπως «σκάσιμο» (απροσδόκητο τέλος), ή bugs (εντελώς απρόβλεπτα λάθη λόγω κακού κώδικα). Εάν το UAT (User Acceptance Test) καταλήγει σε συνεχείς διορθώσεις, τότε η διαδικασία ανάπτυξης κώδικα πάσχει.

Το μότο που θα πρέπει να έχουν οι προγραμματιστές είναι «κάν'το σωστά από την αρχή». Αυτό σημαίνει να δίνεται χρόνος και σκέψη στο σχεδιασμό του προγράμματος, των σεναρίων που θα υλοποιηθούν και στην ορθή υλοποίηση του κώδικα. Πολλοί πιστεύουν ότι το «κάν'το σωστά από την αρχή» σημαίνει ότι μια φορά φτιάχνεται κάτι και δεν πρέπει να μεταβάλλεται. Αυτή είναι διαστρεβλωμένη και επικίνδυνη ερμηνεία. Το πρόγραμμα, η εφαρμογή είναι κάτι «ζωντανό» που εξελίσσεται στην πάροδο του χρόνου. Άρα θα πρέπει να είναι έτσι δομημένο ώστε να είναι ευέλικτο και προσαρμόσιμο στις αλλαγές και να

μπορεί να καλύπτει εύκολα τις νέες, μελλοντικές ανάγκες. Όταν προστίθεται νέα λειτουργικότητα θα πρέπει ο κώδικας που γράφεται να είναι λιτός και απλός. Η πολυπλοκότητα είναι εχθρός του καλού προγραμματισμού και προκαλεί τεράστια προβλήματα στη συντήρηση των προγραμμάτων.

Μία ακόμη κακή τακτική που πρέπει να αποφεύγεται είναι η επαναληψιμότητα του κώδικα (duplication). Εάν μια λειτουργία επαναλαμβάνεται σε κάποια προγράμματα, αυτή θα πρέπει να «χτίζεται» σε κάποιο αντικείμενο (π.χ. Function Module), ώστε να καλείται αυτό το ένα αντικείμενο από όλα τα προγράμματα. Με αυτόν τον τρόπο σε επόμενη αλλαγή της λειτουργίας, θα αλλαχτεί μόνο το συγκεκριμένο αντικείμενο, αντί όλων των προγραμμάτων. Είναι προφανές ότι αυτή η προγραμματιστική κουλτούρα είναι απολύτως και κυριολεκτικά Lean.

#### **4.4 Δημιουργία Γνώσης.**

Όπως έχουμε αναφέρει η λογική του Waterfall στο SW, αλλά και γενικότερα στην ανάπτυξη νέου προϊόντος, δεν είναι αποτελεσματική. Μια βασική αιτία αυτού είναι ότι οι προδιαγραφές δίνονται πολύ νωρίς στην αρχή του project και δεν «επιτρέπεται» να ξεφύγει κανείς από αυτές ή να γίνουν σημαντικές μεταβολές. Οι νέες πληροφορίες και τα δεδομένα που πιθανά αλλάζουν το έργο είναι ανεπιθύμητες. Το αποτέλεσμα συνήθως είναι κάτι που ναι μεν έχει υπογράψει ο πελάτης, αλλά δεν τον ικανοποιεί πλήρως ή είναι πλέον «εκτός αγοράς». Στο Lean όμως που η βασική παράμετρος είναι η προστιθέμενη αξία για τον πελάτη, οι νέες πληροφορίες σε σχέση με τις απαιτήσεις, τις προδιαγραφές και τα αποτελέσματα, είναι απαραίτητες και καλοδεχούμενες, αλλά με τέτοιο τρόπο ώστε να ενσωματωθούν ομαλά στο project και να επιταχύνουν την υλοποίηση αντί να την καθυστερήσουν.

Στον αρχικό σχεδιασμό του συστήματος συνυπολογίζονται οι έως τότε γνωστές προδιαγραφές και απαιτήσεις. Ο λεπτομερειακός όμως σχεδιασμός του SW, γίνεται κατά τη δημιουργία του κώδικα, διότι η δυσκολία και η πολυπλοκότητά διαπιστώνονται σε αυτή τη φάση. Κατά τη δημιουργία του κώδικα(του νέου προϊόντος), ανακύπτουν ερωτήματα τα οποία οδηγούν σε νέα δεδομένα από τους χρήστες(feedback), τα οποία πρέπει να συμπεριληφθούν στο νέο σύστημα.

Ο MacCormack (2001) δίνει μερικές συμβουλές για την επιτυχημένη εφαρμογή των παραπάνω:



- Όσο πιο σύντομα γίνεται, να δοθεί στους χρήστες για αξιολόγηση και feedback ένα παραδοτέο(προϊόν/κώδικας) που θα περιλαμβάνει τη minimum λειτουργικότητα, σύμφωνα με τις υπάρχουσες έως τότε προδιαγραφές.
- Άμεσο/Καθημερινό feedback από τους χρήστες και αντίστοιχα άμεση/καθημερινή ενημέρωση του προϊόντος/κώδικα.
- Η ομάδα πρέπει να έχει την εμπειρία ώστε να πάρει τις σωστές αποφάσεις, σε σχέση με το τι επιπλέον υλοποιείται και τι όχι
- Ο σχεδιασμός του συστήματος πρέπει να γίνει με τρόπο που να είναι εύκολη η προσθήκη αλλαγών/διορθώσεων (modular αρχιτεκτονική).

Όλα τα παραπάνω έχουν ως σκοπό τη δημιουργία γνώσης για το συγκεκριμένο προϊόν (SW ή άλλο). Οι επιτυχημένες εταιρείες στη δημιουργία νέων προϊόντων και SW, αυτή τη γνώση που δημιουργούν μέσα από τον πειραματισμό, έχουν την ικανότητα μέσω των διαδικασιών τους να την κάνουν **κτήμα του οργανισμού**. Ο στόχος είναι η γνώση (tacit knowledge) που παράγεται κατά τη δημιουργία ενός νέου προϊόντος(SW ή άλλο), να μη χαθεί, αλλά να χρησιμοποιηθεί στο μέλλον κατά την ανάπτυξη νέων προϊόντων. Με αυτή τη λογική η εταιρεία αποκτάει ανταγωνιστικό πλεονέκτημα, διότι δε χρειάζεται να «ανακαλύπτει τον τροχό» σε κάθε νέο προϊόν, πετυχαίνει καλύτερους χρόνους ανάπτυξης, τα προϊόντα βγαίνουν γρήγορα στην αγορά, έχει μειωμένα κόστη και φυσικά προσθέτει αξία στον πελάτη.

#### **4.5 Αναβολή Δέσμευσης (Defer Commitment).**

Η έννοια της «αναβολής» έχει να κάνει με εκείνες τις σημαντικές αποφάσεις στην πορεία ενός έργου, που δεν θα μπορούν να αντιστραφούν στη συνέχεια εάν προκύψει κάποιον νέο δεδομένο. Άρα είναι φρόνιμο τέτοιου είδους αποφάσεις να λαμβάνονται κυριολεκτικά «την τελευταία στιγμή», αφού μόνο τότε υπάρχουν όλες οι πληροφορίες για να ληφθεί η απόφαση.

Στην πρώιμη φάση ανάπτυξης του SW, όταν υλοποιούνται οι αρχικά δοθείσες προδιαγραφές, θα πρέπει να αποφεύγονται αποφάσεις που «κλειδώνουν» το σύστημα, και δε θα είναι εύκολο να αλλάξουν στην πορεία. Στο SW θα πρέπει να υπάρχουν σημεία στον κώδικα ή παραμετροποίηση τέτοια που να αφήνουν περιθώρια αλλαγών. Βέβαια η επιλογή των σημείων αυτών και οι αποφάσεις για το τι πρέπει να «κλειδώσει» και τι όχι, απαιτούν την ανάλογη εμπειρία.

Πολλοί προτιμούν να αντιμετωπίζουν τα δύσκολα προβλήματα από την αρχή ώστε να είναι πιο εύκολη η υλοποίηση στη συνέχεια. Σε περιβάλλοντα όμως που κυριαρχεί η αβεβαιότητα και η πολυπλοκότητα όπως του SW Development αλλά και τα επιχειρηματικά, η καλύτερη τακτική είναι ο πειραματισμός με εναλλακτικές λύσεις αφήνοντας τις σημαντικές επιλογές ελεύθερες έως ότου πρέπει να παρθεί η απόφαση.

Στις επιχειρήσεις πολλές φορές επικρατεί η αντίληψη ότι ο σχεδιασμός(planning) αποτελεί και δέσμευση(commitment). Σύμφωνα με το στρατηγό και μετέπειτα Πρόεδρο των ΗΠΑ, Dwight Eisenhower : «Κατά την προετοιμασία για τη μάχη πάντα διαπιστώνω ότι τα σχέδια είναι άχρηστα, αλλά ο σχεδιασμός αναντικατάστατος». Ουσιαστικά, με τη γνωστή αυτή ρήση γίνεται αντιληπτή η διαφορά ανάμεσα στο σχεδιασμό και τη δέσμευση. Ο σχεδιασμός είναι μια εξαιρετικά σημαντική λειτουργία για τον οργανισμό, αναπτύσσει τα αντανακλαστικά του και είναι απαραίτητος για τη δημιουργία της αρχιτεκτονικής ενός πολύπλοκου συστήματος στο υψηλό επίπεδο ( high-level architectural design).

Τα παραγόμενα σχέδια από την άλλη, είναι μάλλον υπερτιμημένα. Τα σχέδια πρέπει να έχουν τη δυνατότητα να αλλάζουν εύκολα σε ένα ευμετάβλητο περιβάλλον, όπως είναι το επιχειρηματικό, και να ανταποκρίνονται στις νέες συνθήκες. Αν ένας οργανισμός έχει την αντίληψη ότι από τη στιγμή που καθορίστηκε ένα πλάνο/σχέδιο, αυτό το σχέδιο απαγορεύεται να αλλάξει, αυτός ο οργανισμός δε θα κρατήσει για πολύ. (Taiichi Ohno, 2001)

Από τα παραπάνω αντιλαμβανόμαστε ότι η καλύτερη τακτική τόσο στο SW Development όσο και στις επιχειρήσεις είναι να μην γίνεται σχεδιασμός και μάλιστα λεπτομερής, αλλά να υπάρχει η ευελιξία της αλλαγής των σχεδίων. Δεν πρέπει να υποκύπτουμε στο μύθο ότι ο σχεδιασμός αποτελεί και δέσμευση.

#### **4.6 Ταχεία Παράδοση (Deliver Fast).**

Στα τέλη του 1980 οι Ιαπωνικές εταιρείες έβγαζαν στην αγορά ένα νέο μοντέλο αυτοκινήτου σε 46 μήνες με 1,7 εκατομμύρια ανθρωποώρες, ενώ οι Αμερικανικές εταιρείες σε 60 μήνες με 3,1 εκατομμύρια ανθρωποώρες.

Οι εταιρείες οι οποίες ανταγωνίζονται με κριτήριο την ταχύτητα συχνά έχουν μεγάλο κοστολογικό πλεονέκτημα σε σχέση με τους ανταγωνιστές: έχουν καταφέρει να εξαλείψουν τη σπατάλη η οποία έχει κόστος. Επίσης έχουν πετύχει πολύ χαμηλά ποσοστά ελαττωματικών προϊόντων. Η επαναλαμβανόμενη και αξιόπιστη ταχύτητα είναι αδύνατη χωρίς εξαιρετική ποιότητα. Επίσης έχουν την κουλτούρα και την ικανότητα να πειραματιστούν με νέες ιδέες κατά την ανάπτυξη των προϊόντων.

Στη βιομηχανία του SW Development επικρατεί η αντίληψη ότι για να έχει καλή ποιότητα το προϊόν, θα πρέπει να γίνονται αργά και προσεκτικά βήματα. Όταν όμως μια βιομηχανία θέτει έναν τέτοιο συμβιβασμό, η εταιρεία εκείνη που θα τον σπάσει θα αποκτήσει τεράστιο ανταγωνιστικό πλεονέκτημα. Η Google είναι ένα χαρακτηριστικό παράδειγμα τέτοιας εταιρείας που δημιουργεί και παραδίδει πολύ ποιοτικό software με πολύ συχνό ρυθμό. Οι εταιρείες που ασχολούνται με το SW και συνεχίζουν να πιστεύουν ότι η ταχύτητα και η ποιότητα είναι ασυμβίβαστες, θα αντιμετωπίσουν σύντομα ζήτημα επιβίωσης.

Βέβαια λέγοντας ταχεία παράδοση, δεν εννοούμε βιαστικά και απρόσεκτα. Μια γρήγορη SW Development ομάδα πρέπει να έχει εξαιρετικά αντανακλαστικά και κουλτούρα Stop-The-Line. Ο προφανής λόγος είναι ότι δεν μπορεί να διατηρηθεί η μεγάλη ταχύτητα παράδοσης χωρίς ποιοτικό SW. Τα χαρακτηριστικά μιας τέτοιας ομάδας είναι ο ενθουσιασμός για αυτό που κάνουν, η δέσμευση στην αποστολή και στους στόχους της εταιρείας, η συνεργασιμότητα μεταξύ τους και φυσικά η γνώση του αντικειμένου. Τέτοιοι οργανισμοί έχουν στόχο να αναπτύσσουν τους ανθρώπους τους ώστε να παίρνουν πρωτοβουλίες και να λύνουν τα προβλήματα γρήγορα αλλά και ποιοτικά.

Υπάρχουν δύο τρόποι για να πετύχει μια εταιρεία υψηλή ποιότητα προϊόντων. Αργά και προσεκτικά(παραδοσιακά) ή με την ανάπτυξη ανθρώπινου δυναμικού το οποίο θα βελτιώνει συνεχώς τις διαδικασίες του, θα αυξάνει την ποιότητα των προϊόντων και θα παραδίδει ακόμα γρηγορότερα(lean).

#### **4.7 Σεβασμός στους Ανθρώπους**

Όπως είχαμε δει στην Ιστορική Αναδρομή, ο Frank Gilbreth και ο Frederick Taylor είχαν ασχοληθεί με τη βελτίωση των διαδικασιών στις βιομηχανικές εργασίες. Βέβαια η βελτίωση αυτή δεν λάμβανε ιδιαίτερα υπόψη την ψυχολογία και την προσωπικότητα του εργατή.

Προσπαθούσαν να βρουν τον έναν καλύτερο τρόπο εκτέλεσης της εργασίας, πιστεύοντας ακράδαντα ότι υπάρχει μόνο ένας μοναδικός βέλτιστος τρόπος(There is One Best Way). Η πραγματικότητα όμως δείχνει ότι πρέπει να καταρριφθεί αυτός ο μύθος.

Ο σεβασμός στους ανθρώπους σημαίνει ότι αυτοί που κάνουν τη δουλειά, έχουν πολύ σημαντικό ρόλο κι αυτό πρέπει να αναγνωρίζεται από την εταιρεία, δίνοντάς τους τη δυνατότητα να αποφασίζουν μόνοι τους για το πώς θα κάνουν τη δουλειά τους, καθώς και για τη βελτίωση των διαδικασιών τους. Κάθε διαδικασία μπορεί να βελτιωθεί, αρκεί να υπάρχει διάθεση, καθοδήγηση και εμπιστοσύνη σε αυτούς που κάνουν τη δουλειά.

Ειδικά οι «εργάτες γνώσης» όπως είναι οι προγραμματιστές SW δε χρειάζονται κάποιον που να τους δίνει συνεχείς οδηγίες για το πώς και τι θα υλοποιήσουν. Αυτό που χρειάζονται είναι γενικές οδηγίες, λογικούς στόχους, υποστήριξη κι ένα περιβάλλον που θα σέβεται την προσωπικότητά τους και θα δείχνει εμπιστοσύνη στην ικανότητά τους να αυτο-οργανωθούν και να πετύχουν τους στόχους.

Κάθε εταιρεία που θέλει να πετύχει ανταγωνιστικό πλεονέκτημα σε έναν τομέα, πρέπει να αναπτύσσει το κατάλληλα εξειδικευμένο προσωπικό. Οι εταιρείες που αγοράζουν την εξειδίκευση, θα διαπιστώσουν ότι το ίδιο ακριβώς μπορούν να κάνουν και οι ανταγωνιστές. Αυτό είναι ιδιαίτερα σημαντικό για τις εταιρείες που έχουν SW Development τμήματα. Για να πετύχουν στόχους και μακροημέρευση οφείλουν να αναπτύσσουν τη δική τους ομάδα που θα τη σέβονται, θα τη βλέπουν ως επένδυση και όχι ως κόστος.

Όσον αφορά στην ηγεσία, είναι εξίσου σημαντικό για μια εταιρεία να αναπτύσσει εσωτερικά καλούς ηγέτες και να εξασφαλίζει ότι οι ομάδες θα έχουν μια ηγεσία η οποία θα εμπνέει σοβαρούς, σκεπτόμενους και δεσμευμένους στο στόχο ανθρώπους.

#### **4.8 Βελτιστοποίηση του Συνόλου (Optimize the Whole).**

Ένας Lean SW οργανισμός οφείλει να βελτιστοποιεί ολόκληρο το value stream (End-to-End), από τη στιγμή που θα παραληφθεί η παραγγελία του πελάτη έως τη στιγμή που το SW έχει παραδοθεί. Αν προσπαθήσει να βελτιστοποιήσει μικρότερο τμήμα της αλυσίδας, τότε θα υπάρξει πρόβλημα ή τουλάχιστον δεν θα υπάρξει η αναμενόμενη βελτίωση. Όταν στο value stream υπάρχει μεγάλη καθυστέρηση, αυτή συνήθως οφείλεται σε μετακύλιση της ευθύνης σε άλλο τμήμα (hand-off), χωρίς την ύπαρξη ενός υπευθύνου που θα ασχολείται με την πραγματική ανάγκη του πελάτη.

Ο μύθος που πρέπει να καταρριφθεί είναι η τμηματική βελτιστοποίηση (Optimize by Decomposition). Πολλοί πιστεύουν ότι αν βελτιστοποιηθούν τα επιμέρους τμήματα του value stream, θα υπάρξει βελτίωση και συνολικά. Όμως δεν ισχύει κάτι τέτοιο. Αν το value stream διασπαστεί σε τμήματα, ακόμα κι αν οι μετρήσεις απόδοσης των επιμέρους τμημάτων είναι βελτιωμένες, η εμπειρία δείχνει πως η συνολική απόδοση δεν θα είναι.

Η λύση σε αυτές τις περιπτώσεις αντί της μέτρησης της απόδοσης των επιμέρους τμημάτων, είναι η εύρεση μιας μέτρησης σε ψηλότερο επίπεδο και μείωση των διαφόρων μετρητών (measurements). Π.χ. για την απόδοση των SW projects παραδοσιακά χρησιμοποιούνται μετρητές βασισμένοι στο κόστος, στον προγραμματισμό (schedule), στους στόχους του scope (scope targets), στην ποιότητα, στην ικανοποίηση του πελάτη κλπ. Αντί όλων των επιμέρους μετρητών θα μπορούσε να χρησιμοποιηθεί κάποιος άλλος σε υψηλότερο επίπεδο. Μία πρόταση θα ήταν ένας ROI (Return On Investment) για τη διαμόρφωση και μέτρηση του οποίου θα έπρεπε να συμμετάσχουν τόσο ο Business Owner του project όσο και το SW Department.

## ΚΕΦΑΛΑΙΟ 5. ΔΗΜΙΟΥΡΓΙΑ ΑΞΙΑΣ ΓΙΑ ΤΟΝ ΠΕΛΑΤΗ

Στο παρόν κεφάλαιο θα εστιάσουμε στη Δημιουργία αξίας για τον πελάτη/καταναλωτή σύμφωνα με τη Lean φιλοσοφία και θα αναφερθούμε στην έννοια Πελάτης ιδιαίτερα στο SW development.

### 5.1 Lean Consumption.

Ο όρος **Lean Consumption**, ο οποίος χρησιμοποιήθηκε από τους James P.Womack και Daniel T. Jones στο άρθρο τους Lean Consumption HBR 2005, αφορά στην αύξηση της αξίας για τον πελάτη, μέσω της επίλυσης προβλημάτων που αντιμετωπίζει στα στάδια μετά την αγορά του προϊόντος/υπηρεσίας.

Οι διαδικασίες που έχει ο πάροχος του προϊόντος/υπηρεσίας, πρέπει να αποβλέπουν στην εξάλειψη της σπατάλης μέσα στον οργανισμό του και ταυτόχρονα στην εξάλειψη της σπατάλης που βιώνει ο πελάτης.(π.χ. αργοπορίες, λάθη κατά την παράδοση, κλπ. οδηγούν τον πελάτη σε χάσιμο χρόνου, χρημάτων, αλλά και σε κακή ψυχολογία).

Με άλλα λόγια η καλή ποιότητα του προϊόντος ή της υπηρεσίας δεν είναι πλέον αρκετή για τον πελάτη. Τον ενδιαφέρει εξίσου ο τρόπος με τον οποίο του παρέχεται το προϊόν/υπηρεσία καθώς και η ευκολία χρήσης του.

Για να δημιουργήσουμε αξία για τον πελάτη, πρέπει να αντιληφθούμε την αξία του προϊόντος/υπηρεσίας που παρέχουμε, από την **οπτική του πελάτη**. Αυτή ακριβώς είναι η φιλοσοφία του Lean όσον αφορά στην εξυπηρέτηση του πελάτη, και έχει να κάνει με όλη την αλυσίδα αξίας όπως την αντιλαμβάνεται ο πελάτης/καταναλωτής. Η αλυσίδα αυτή δε σταματάει τη στιγμή της πώλησης, αλλά επεκτείνεται έως την πλήρη χρήση του προϊόντος/υπηρεσίας. Είναι η **End-to-End** αλυσίδα από τη στιγμή που ο πελάτης εκδηλώνει επιθυμία επίλυσης ενός προβλήματος έως τη **ριζική** επίλυση αυτού του προβλήματος, με τη **μέγιστη ταχύτητα και χωρίς προβλήματα** για εκείνον.

Για να επιτευχθεί ο στόχος αυτός, πρέπει να σκεφτόμαστε την κατανάλωση όχι ως μεμονωμένο γεγονός (πώληση), αλλά ως **συνεχή διαδικασία** που συνδέει όλες τις απαραίτητες υπηρεσίες και αγαθά ώστε να επιλύσει το πραγματικό πρόβλημα του καταναλωτή.

Στη συνέχεια θα παραθέσουμε τις βασικές αρχές του Lean Consumption κατά Womack-Jones (2005) :

**a. Ολοκληρωτική επίλυση του προβλήματος εξασφαλίζοντας ότι όλα τα προϊόντα και υπηρεσίες (της δοθείσας λύσης) λειτουργούν και συνεργάζονται μεταξύ τους**

Παράδειγμα: η μη καλή επικοινωνία εκτυπωτή – software με αποτέλεσμα να μην πραγματοποιείται η εκτύπωση. Το πρόβλημα του πελάτη/χρήστη είναι « ο εκτυπωτής δεν τυπώνει». Ο τεχνικός του εκτυπωτή που έχει αρχικά αναλάβει την επίλυση «βλέπει» μόνο τη δική του μεριά, δηλώνει ότι ο εκτυπωτής είναι λειτουργικός, δεν έχει πρόβλημα και **παραπέμπει** τον πελάτη/χρήστη στο SW department ώστε να δοθεί εκεί η επίλυση. Η ορθή (lean) προσέγγιση είναι η **άμεση συνεργασία ανάμεσα στα δύο τμήματα** (τεχνικό-software), χωρίς την παρέμβαση του πελάτη/χρήστη (τουλάχιστον έως την εύρεση της λύσης και τον έλεγχό της). Δεν πρέπει ο πελάτης/χρήστης να αναλώνεται στο να βρει τον υπεύθυνο του προβλήματός του. Εκτός όμως από τη συνεργασία των τμημάτων, θα πρέπει να δοθεί **οριστική και βιώσιμη λύση** στο πρόβλημα, και όχι πυροσβεστική αντιμετώπιση.

**b. Μη σπαταλάς το χρόνο του πελάτη**

Η έγκαιρη και ολοκληρωμένη παράδοση προϊόντος/υπηρεσίας είναι ζητούμενο και θεωρείται δεδομένο, αλλά στην πραγματικότητα δεν ισχύει κάτι τέτοιο. Είναι πολλές οι φορές στις οποίες ο πελάτης/χρήστης πρέπει να υπενθυμίζει, να κυνηγάει την επίλυση του προβλήματός του, να αντιμετωπίζει καθυστερήσεις κατά την παράδοση ή/και να παραλαμβάνει ελαττωματικό προϊόν/υπηρεσία. Το αποτέλεσμα είναι να πρέπει να ξαναασχοληθεί με το θέμα, διότι είναι σίγουρο ότι θα το αντιμετωπίσει ξανά.

Παράδειγμα : Ασυμφωνία σε κλείσιμο μήνα στο Λογιστήριο. Η πυροσβεστική αντιμετώπιση είναι να βρίσκονται οι λογιστικές εγγραφές που προκαλούν την ασυμφωνία και να γίνονται διορθωτικές κινήσεις. Ο πελάτης/χρήστης χάνει το χρόνο του τόσο στη σύνταξη του προβλήματος ώστε να το αποστέλλει στο IT κάθε μήνα, όσο και στο να δημιουργήσει τις διορθωτικές εγγραφές που θα του προτείνει το IT, αφού βρεθούν τα λάθη. Η ορθή (lean) προσέγγιση είναι η εύρεση του προβλήματος στην πηγή από το IT σε συνεργασία με το χρήστη. Για ποιο λόγο δημιουργούνται οι λανθασμένες εγγραφές και πώς αυτό θα διορθωθεί.

**c. Δώσε στον πελάτη ακριβώς αυτό που ζήτησε**

Ειδικά στο SW development αυτή η αρχή σπάνια ικανοποιείται. Συνήθως δημιουργούνται πολλές (και άχρηστες) λειτουργίες, ως πρόβλεψη των απαιτήσεων του πελάτη/χρήστη. Με τον τρόπο αυτό όμως δημιουργούνται προγράμματα με πολλές γραμμές

αχρησιμοποίητου στην πραγματικότητα κώδικα. Βεβαίως οι παραπανίσιες αυτές γραμμές κώδικα μεταφράζονται σε χαμένο χρόνο, παραπάνω κόπο και πολυπλοκότητα. Δηλ. είναι ξεκάθαρη σπατάλη σύμφωνα με τη Lean σκέψη, αλλά και με την κοινή λογική.

Παράδειγμα: Για νομικούς λόγους απαιτήθηκε η δημιουργία Χρεωστικών και Πιστωτικών σημειωμάτων σε εκτυπωτική εφαρμογή η οποία δεν είχε δημιουργηθεί για το σκοπό αυτό.

Παράλληλα έγινε και η «πρόβλεψη» να μπορεί να εκδίδεται και Ειδικό Ακυρωτικό Στοιχείο. Τελικά αποφασίστηκε ότι η συγκεκριμένη πρόβλεψη της έκδοσης ακυρωτικών δεν ήταν τόσο μεγάλης σημασίας και ότι όλες οι περιπτώσεις μπορούσαν να καλυφθούν μόνο με Χρεωστικά-Πιστωτικά.

#### **d. Δώσε στον πελάτη ακριβώς αυτό που ζήτησε, εκεί που το θέλει**

Η συγκεκριμένη αρχή αφορά στην ευκολία με την οποία παρέχεται το προϊόν στον πελάτη σε συνδυασμό με την τιμή. Συνήθως η καλή υπηρεσία κοστίζει ακριβά.

Η lean σκέψη εστιάζει στο να λάβει ο πελάτης το ποιοτικό προϊόν/υπηρεσία σε ανταγωνιστική τιμή.

#### **e. Δώσε στον πελάτη ακριβώς αυτό που ζήτησε, εκεί που το θέλει, όταν το θέλει**

Το κεντρικό σημείο της Lean σκέψης είναι, όπως έχουμε δει, η πλήρης ικανοποίηση του πελάτη. Με βάση αυτό πρέπει εκτός των παραπάνω να λαμβάνεται υπ'όψιν ο χρόνος στον οποίο ο πελάτης θέλει να παραλάβει και ειδικά σε συνάρτηση με την τιμή. Είναι πολύ λογικό για τον πελάτη να ζητήσει καλύτερη τιμή εάν θελήσει να αγοράσει κάτι μετά από κάποιο διάστημα, αλλά το κανονίζει εκ των προτέρων. Βέβαια τέτοιου είδους εξυπηρέτηση απαιτεί εξαιρετική συνεργασία ανάμεσα στους συνεργαζόμενους προμηθευτές για τη δημιουργία του προϊόντος/υπηρεσίας.

Παράδειγμα: Τουριστικά πακέτα που κλείνονται από τον προηγούμενο χρόνο, με καλύτερη τιμή για τον πελάτη και καλύτερη ποιότητα. Θα πρέπει να υπάρξει συνεργασία όλων των προμηθευτών. (ταξιδιωτικό γραφείο, ξενοδοχείο, αεροπορικές εταιρείες κλπ.)

Και καταλήγουμε στην τελευταία αρχή, η οποία ουσιαστικά αφορά όλες τις προηγούμενες.

#### **f. Ενοποίησε λύσεις ώστε να μειώνεται ο χρόνος και η ταλαιπωρία του πελάτη**

Ο στόχος είναι να ανακαλύπτονται νέοι τρόποι για την πλήρη εξυπηρέτηση του πελάτη με όσο γίνεται λιγότερα προβλήματα και αργοπορία για τον πελάτη/χρήστη.



Αυτό σημαίνει ότι θα πρέπει να υπάρξουν κανάλια επικοινωνίας ανάμεσα σε προμηθευτές και πελάτες, ώστε να σχεδιάζονται οι λύσεις και από τα δύο μέρη όσο γίνεται πιο αποδοτικά και για τα δύο μέρη. Και οι πελάτες πρέπει να είναι ικανοποιημένοι, αλλά και οι πάροχοι να έχουν κέρδος.

Παράδειγμα: Λόγω της μεγάλης πίεσης χρόνου που αντιμετωπίζουν τις περισσότερες φορές τα τμήματα μιας εταιρείας, συμβαίνει πολλές φορές να έχει υλοποιηθεί η λύση η οποία έχει ζητηθεί από το IT, αλλά να μην προλαβαίνουν οι χρήστες να ελέγξουν. Θεωρώ ότι η λύση θα ήταν, να συνεννοηθούν τα δύο μέρη (IT – χρήστες) ώστε να βρεθεί ο κατάλληλος χρόνος, έτσι ώστε μόλις το IT ολοκληρώσει την υλοποίηση, οι χρήστες να έχουν το χρόνο για έλεγχο.

## **5.2 Η έννοια «Πελάτης» στο εσωτερικό SW Development.**

Ειδικά στο SW development οι πελάτες μπορεί να είναι εξωτερικοί, π.χ. για ένα software house ή εσωτερικοί π.χ. οι συνάδελφοι των άλλων τμημάτων σε μια εταιρεία.

Σύμφωνα με τον MacGibbon και τους συνεργάτες του (2006), τα εσωτερικά τμήματα IT SW Development θα πρέπει να λειτουργούν με τον ίδιο τρόπο που λειτουργούν τα SW Houses, όσον αφορά την αντιμετώπιση των «πελατών» τους, δηλ. των άλλων συναδέλφων τους. :

1. Τα SW Houses, ως επιχειρήσεις, πραγματοποιούν έρευνες αγοράς, ώστε να αντιληφθούν τις ανάγκες των πελατών τους και να τις ικανοποιήσουν. Αν δεν πράξουν με αυτόν τον τρόπο, προφανώς θα χάσουν μερίδιο, πελάτες, ακόμα και την ίδια την επιχείρησή τους. Στα IT departments όμως ξεχνάμε την έννοια της έρευνας αγοράς, και παραδίδουμε υπηρεσίες (προγράμματα) που δεν ανταποκρίνονται στις ανάγκες του πελάτη μας (business).
2. Τα SW Houses επειδή είναι επιχειρήσεις και στοχεύουν στο κέρδος, προσπαθούν εκτός από την αύξηση των εσόδων, την ταυτόχρονη μείωση των εξόδων. Σχεδιάζουν τα προϊόντα τους (SW), με τέτοιο τρόπο ώστε να είναι ευέλικτα κατά τη δημιουργία αλλά και την αλλαγή/επέκταση/ενίσχυση/βελτίωση, έτσι ώστε να έχουν όσο γίνεται λιγότερο κόστος. Πολύ συχνά στα IT Departments γίνεται «κατάχρηση» στη δημιουργία προγραμμάτων με πολλές λειτουργίες, παρόλο που είναι ακριβές και δύσκολες, τόσο κατά τη δημιουργία τους, όσο και σε πιθανές βελτιώσεις/ενισχύσεις.

3. Τα SW Houses αντιλαμβάνονται πλήρως ότι για να έχουν συνέχεια και μακροζωία, θα πρέπει να έχουν «ευτυχείς» πελάτες. Τα προϊόντα τους πρέπει να είναι σταθερά, να ανταποκρίνονται πλήρως στις ανάγκες του πελάτη και να μπορούν εύκολα να εξελιχθούν. Στα IT Departments υπάρχει σε μεγάλο βαθμό (απίστευτα μεγάλο) η πεποίθηση ότι η επιτυχία των προϊόντων τους (προγραμμάτων) οφείλεται στις ορθές προδιαγραφές του business, κι ότι το business είναι υπεύθυνο για το εκάστοτε project. Προφανώς κάτι τέτοιο έχει λογική βάση. Εννοείται ότι το Business που έχει την απαίτηση έχει την ευθύνη των προδιαγραφών, του ελέγχου και της επιτυχίας ή αποτυχίας του εγχειρήματος, αλλά και το IT πρέπει να αντιληφθεί ότι το προϊόν του δεν είναι επιτυχημένο εάν δεν συμβάλλει καθοριστικά στην επιτυχία του Business.

### 5.3 Συνεργασία Business – IT.

Η δημιουργία ενός προϊόντος SW γίνεται συνήθως μέσω της διαδικασίας Project με συγκεκριμένο score, blueprint, χρονοδιάγραμμα κλπ.

Όμως όταν το IT είναι ενσωματωμένο σε μια εταιρεία, πολλά στοιχεία της διαδικασίας που ακολουθούνται στο Project Management θα μπορούσαν να παραλειφθούν. Σε τέτοιες περιπτώσεις δεν είναι απαραίτητο να υπάρχουν πολλές Project Management διαδικασίες, όπως εμπειριστωμένο και λεπτομερές Project Score από την αρχή του έργου, δε χρειάζονται sign-off, ούτε λεπτομερής έλεγχος του score μήπως ζητηθεί κάτι extra. Επίσης μπορεί να υπάρχει ευελιξία, ώστε να ζητούνται και να παραδίνονται τα απαραίτητα με σειρά προτεραιότητας / σημαντικότητας. Πολλές φορές ζητούνται λειτουργίες / δυνατότητες από το business εκ των προτέρων, μόνο και μόνο επειδή υπάρχει ο φόβος ότι εάν δεν ειπωθούν και προδιαγραφούν δεν θα γίνουν ποτέ! Βέβαια επειδή ακριβώς δεν μπορούν να υλοποιηθούν όλα παράλληλα και ταυτόχρονα, θα πρέπει να υπάρχει διαβάθμιση των ζητούμενων λειτουργιών και ανάλογη υλοποίησή τους ή όχι.

Η διαβάθμιση αυτή θα μπορούσε να γίνει βάσει ερωτηματολογίων π.χ. οι χρήστες να απαντάνε σε ένα ερωτηματολόγιο που θα «βαθμολογεί» τη λειτουργικότητα ως προς την ικανοποίηση που θα τους προσφέρει ή βάσει της σημασίας που της προσδίδουν. Ο στόχος είναι να δημιουργηθούν προϊόντα SW που θα προσφέρουν τη μέγιστη ικανοποίηση στον πελάτη/χρήστη, με λειτουργικότητα που εκείνος θα χρησιμοποιήσει πραγματικά. Όπως έχει ήδη ανφερθεί, πολλές φορές χτίζονται λειτουργικότητες που δε

χρησιμοποιούνται ποτέ, πράγμα που είναι απολύτως αντίθετο με τη Lean φιλοσοφία, διότι είναι κυριολεκτικά σπατάλη. Προφανώς το IT έχει την υποχρέωση να δημιουργεί αξία για τους πελάτες του (business) και να τους ικανοποιεί στο μέγιστο βαθμό. Αυτό όμως δε σημαίνει ότι όλες οι απαιτήσεις των χρηστών πρέπει να υλοποιούνται, ειδικά εάν δεν πρόκειται να χρησιμοποιηθούν.

Το IT και το Business θα πρέπει να χτίσουν μια υγιή σχέση αλληλοσεβασμού και σύμπνοιας προς τον κοινό σκοπό: την ευημερία και μακροημέρευση της εταιρείας.

#### 5.4 Συμπεράσματα

- Ακούμε προσεκτικά τη «φωνή του πελάτη». Λύνουμε ολοκληρωτικά το πρόβλημα και όχι πυροσβεστικά. (**Βρίσκουμε την πραγματική αιτία του προβλήματος**). Χρήση Lean μεθόδου 5-Why. ( Ρωτάμε και ξαναρωτάμε ώστε να φτάσουμε στην πηγή του προβλήματος).
- Χρησιμοποιούμε το SW Development ως προϊόν και όχι ως project. Η ανάπτυξή του είναι δυναμική. Δε χρειάζονται επακριβείς και απόλυτα ξεκάθαρες λεπτομέρειες από την αρχή.
- Σχεδιάζουμε προσεκτικά τη λύση και την αρχιτεκτονική της, ώστε να είναι ευέλικτη σε πιθανές αλλαγές/ενισχύσεις.
- Δημιουργούμε αυτά που πρέπει και δεν φορτώνουμε τον κώδικα με άχρηστες γραμμές και λειτουργικότητες που δε θα χρησιμοποιηθούν.
- Κάνουμε έρευνα ικανοποίησης πελάτη (business), ώστε να μαθαίνουμε από τα λάθη μας και να βελτιώνουμε τις υπηρεσίες μας.

## ΚΕΦΑΛΑΙΟ 6. ΣΠΑΤΑΛΕΣ ΣΤΟ SW DEVELOPMENT

### 6.1 Εισαγωγή.

Όσοι έχουν ασχοληθεί με τη φιλοσοφία του Lean στο manufacturing ή σε άλλο τομέα, γνωρίζουν τις 7 σπατάλες του manufacturing όπως τις έχει κατηγοριοποιήσει ο Shigeo Shingo (1981). Αντίστοιχες σπατάλες, τις οποίες θα αναλύσουμε διεξοδικά, διαπιστώνονται και στο SW. (Mary and Tom Poppendieck, 2007). Επίσης προσθέτουμε και τη βασική σπατάλη της υποαπασχόλησης.

Η κατηγοριοποίηση δεν είναι αυτοσκοπός αλλά βοηθάει στο να γίνεται η σπατάλη ευκολότερα αντιληπτή. Ο πραγματικός σκοπός της εύρεσης(αποκάλυψης) της σπατάλης και της εξάλειψής της, είναι φυσικά η μείωση του κόστους και η αύξηση της αξίας προς τον πελάτη.

Στον παρακάτω πίνακα παρουσιάζεται η αντιστοίχιση ανάμεσα στις σπατάλες της βιομηχανίας και του SW Development.

<b>Manufacturing</b>	<b>SW Development</b>
Αποθέματα (Inventory)	Ημιτελής δουλειά (Partially Done Work)
Υπερπαραγωγή (Over-production)	Υπερβάλλουσα λειτουργικότητα (Extra Features)
Επανεπεξεργασία (Extra Processing)	Επανεύρεση λύσης (Relearning)
Μεταφορές (Transportation)	Μεταφορά εργασίας (Handoffs)
Κίνηση (Motion)	Εναλλαγή εργασίας (Task switching )
Αναμονές (Waiting)	Καθυστερήσεις-Αναμονές (Delays)
Ελαττωματικά (Defects)	Ελαττωματικός κώδικας (Defects)
Underutilization(υποαπασχόληση)	Underutilization(υποαπασχόληση)

Πίνακας 6. Σπατάλες στο SW Development

## 6.2 Ημιτελής δουλειά. (Partially Done Work)

Ο στόχος στο SW είναι η δημιουργία ολοκληρωμένου, ελεγμένου και τεκμηριωμένου κώδικα που μεταφέρεται άμεσα στην παραγωγική διαδικασία. Η ροή αυτή πρέπει να «κυλάει οριζόντια» πολύ γρήγορα και χωρίς διακοπές. Ο μόνος τρόπος για να επιτευχθεί κάτι τέτοιο είναι να σπάει η προγραμματιστική δουλειά σε μικρότερα τμήματα, ώστε να είναι ευκολότερα διαχειρίσιμη.

Στις ημιτελείς εργασίες στο SW περιλαμβάνονται οι παρακάτω :

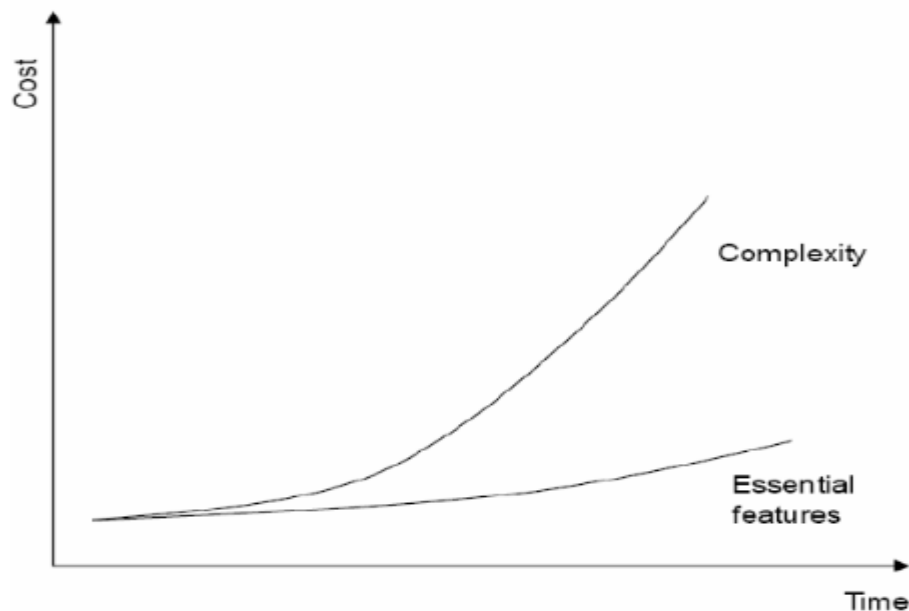
- **Προδιαγραφές/Απαιτήσεις** που μένουν στο στάδιο της καταγραφής και δεν προχωράνε στο επόμενο βήμα της κωδικοποίησης. Όσο πιο πολύ αργεί η υλοποίηση του κώδικα, τόσο πιο πιθανό είναι ότι οι προδιαγραφές θα αλλάξουν συμβαδίζοντας με τις ανάγκες των χρηστών. Όπως είναι προφανές, αυτό είναι ένα σημείο τριβής ανάμεσα στο SW Department και το Business διότι ο καθένας το βλέπει από τη δική του οπτική, όπως έχουμε δει σε προηγούμενο κεφάλαιο.
- **Μη ελεγμένος κώδικας.** Ο κώδικας πρέπει να ελέγχεται άμεσα από τους χρήστες και να μην παραμένει ως εκκρεμότητα προς έλεγχο (User Action).
- **Μη τεκμηριωμένος κώδικας.** Η τεκμηρίωση θα πρέπει να γράφεται κατά τη διάρκεια της συγγραφής του κώδικα ή στο τέλος, ώστε να περιλαμβάνει αυτά που πραγματικά κάνει το πρόγραμμα. Επίσης ο ίδιος ο κώδικας θα πρέπει να είναι έτσι δομημένος και με σύνταξη τέτοια που να είναι εύκολη η «ανάγνωσή» του.
- **Κώδικας που δεν μεταφέρεται στην παραγωγή.** Το επιδιωκόμενο είναι ο ολοκληρωμένος, ελεγμένος, τεκμηριωμένος κώδικας να περάσει στην παραγωγή όσο πιο σύντομα γίνεται. Πολλές φορές αυτό είναι δύσκολο, διότι το νέο software μπορεί να προκαλέσει αναστάτωση στη δουλειά των χρηστών, χωρίς να εννοούμε προβλήματα της εφαρμογής. Πιθανά να πρέπει να «βγουν» οι χρήστες εκτός συστήματος, να πρέπει να σταματήσουν οι εργασίες τους για κάποιο διάστημα έως ότου περαστούν οι αλλαγές κλπ. Παρόλα αυτά η προσπάθεια όλων θα πρέπει να είναι η μεταφορά στην παραγωγή όσο πιο σύντομα γίνεται μετά από την τελική οριστικοποίηση του ελέγχου από τους χρήστες.

### 6.3 Υπερβάλλουσα λειτουργικότητα. (Extra Features)

Ο Taiichi Ohno θεωρούσε ως χειρότερη σπατάλη όλων, την υπερπαραγωγή αποθεμάτων (overproduction). Αντίστοιχα όμως και στο SW η χειρότερη σπατάλη είναι η δημιουργία λειτουργικότητας/λειτουργιών που δεν θα χρησιμοποιηθούν από τον πελάτη/χρήστη. Εάν η λειτουργικότητα δεν τεκμηριώνεται ξεκάθαρα και με οικονομικούς όρους είναι άχρηστος κώδικας και **δεν θα πρέπει να χτίζεται**.

Η παραγωγή άχρηστου κώδικα προκαλεί μεγάλη πολυπλοκότητα. Σύμφωνα με τον Michael George(2004), η πολυπλοκότητα προσομοιάζει στη χοληστερόλη: φράζει τις αρτηρίες του οργανισμού και είναι ο σιωπηλός δολοφόνος κερδών και ανάπτυξης. Η συνταγή εναντίον της πολυπλοκότητας στο SW είναι απλά η συγγραφή **λιγότερου κώδικα**.

Το κόστος συντήρησης της πολυπλοκότητας αυξάνεται εκθετικά σε συνάρτηση με το χρόνο. Ο πολύπλοκος κώδικας είναι εύθραυστος και επιρρεπής σε λάθη, με αποτέλεσμα να καθίσταται ασταθής και αβέβαιη οποιαδήποτε αλλαγή/ενίσχυση πρέπει να υλοποιηθεί.



Σχήμα 10. Κόστος πολυπλοκότητας

Το πρώτο βήμα για τον περιορισμό της πολυπλοκότητας είναι ο περιορισμός εξαρχής των πολλών λειτουργιών. Για κάθε λειτουργικότητα που ζητείται να μπει στον κώδικα, θα πρέπει να αποδεικνύεται η αναγκαιότητά της και μάλιστα με οικονομικούς όρους. Θα πρέπει δηλ. να αποτιμάται το κέρδος που θα αποκομίσει ο χρήστης με τη νέα λειτουργία.

Αυτή η αντιμετώπιση απαιτεί γενναιότητα και αποφασιστικότητα. Είναι δεδομένο ότι θα υπάρξει αντίσταση από τη μεριά των χρηστών στην προσπάθειά τους να προσθέσουν λειτουργικότητα, την οποία βέβαια δε θα μπορούν να τεκμηριώσουν ως υψηλής προτεραιότητας.

Εδώ έρχεται το επόμενο βήμα στην προσπάθεια εναντίον της πολυπλοκότητας κι αυτό είναι η τμηματοποίηση της υλοποίησης. Τα στάδια υλοποίησης του κώδικα πρέπει να καθορίζονται βάσει των λειτουργιών που πρέπει να υλοποιηθούν. Οι λειτουργίες να προτεραιοποιούνται και η υλοποίηση να προχωράει προφανώς από τις λειτουργίες υψηλής προτεραιότητας προς εκείνες με χαμηλότερη προτεραιότητα.

Όταν μιλάμε για υλοποίηση λειτουργιών εννοούμε όλο το flow που περιγράψαμε παραπάνω : ολοκληρωμένο κώδικα, ελεγμένο και αποδεκτό από τους χρήστες, πλήρως τεκμηριωμένο, ο οποίος μεταφέρεται άμεσα στην παραγωγή. Με αυτήν την τακτική υλοποίησης οι χρήστες θα δουλεύουν κάθε νέα λειτουργικότητα πολύ πιο γρήγορα, κι έτσι θα έχουν τη δυνατότητα να αντιληφθούν πραγματικά πώς δουλεύει η εφαρμογή και να ζητήσουν ή όχι αλλαγές/βελτιώσεις.

Μια άλλη σημαντική πολιτική που πρέπει να ακολουθείται είναι η μη αυτοματοποίηση της πολυπλοκότητας. Πολλές φορές ζητείται, και δυστυχώς γίνεται, σύνθετες και περίπλοκες διαδικασίες να αυτοματοποιούνται για μεγαλύτερη ταχύτητα. Σε τέτοιες περιπτώσεις πρέπει οπωσδήποτε να ελέγχεται η διαδικασία προς αυτοματισμό, ώστε ει δυνατόν να απλοποιείται. Μόνο τότε θα μπορούν να βρεθούν τα ακριβή και ορθά σημεία αυτοματοποίησης.

#### **6.4 Επανεύρεση λύσης (Relearning).**

Στο SW Development αλλά και γενικότερα σε τομείς που παράγεται γνώση σε οποιαδήποτε μορφή, είναι πολύ σημαντική η διατήρησή της και η εύκολη ανεύρεσή της. Το πλέον σημαντικό όμως αλλά και πιο δύσκολο είναι η **μη άγνοια** της υπάρχουσας γνώσης.

Είναι συχνό το φαινόμενο να «ανακαλύπτουμε ξανά τον τροχό», δηλαδή να αναζητείται η λύση που είχε δοθεί σε κάποιο πρόβλημα και λόγω μη ύπαρξης τεκμηρίωσης ή δύσκολης ανεύρεσής της, να πρέπει να ανακαλυφθεί ή να υλοποιηθεί ξανά.

Ακόμη όμως και να υπάρχει επαρκής και προσβάσιμη πληροφόρηση για κάποιο ζήτημα, είναι πιθανότατο ο ενδιαφερόμενος να έχει άγνοια της ύπαρξής της, με αποτέλεσμα να ξαναδουλέψει και να επανεφεύρει τη λύση.

Ειδικά στο SW Development τα παραπάνω οδηγούν σε υπερπαραγωγή (overproduction), αφού δημιουργείται(ξανά) SW που ικανοποιεί τις ίδιες ανάγκες.

### **6.5 Μεταφορά εργασίας (Handoffs).**

Το λεγόμενο handoff, η παράδοση ημιτελούς δουλειάς σε επόμενο συνάδελφο ώστε να την προχωρήσει, μοιραία οδηγεί σε απώλεια γνώσης. Αναφερόμαστε στη «σιωπηρή γνώση» (tacit knowledge), που είναι εκείνη η γνώση η οποία δεν μπορεί να καταγραφεί εύκολα σε documentations αλλά παραμένει κτήμα εκείνου που τη δημιούργησε.

Ο στόχος είναι η γνώση αυτή, η οποία είναι δύσκολο να επικοινωνηθεί, να μη σπαταλιέται. Μέθοδοι αποφυγής αυτή της διαρροής γνώσης θα μπορούσαν να είναι οι παρακάτω :

- Μείωση του αριθμού των handoffs. Είναι αποδοτικότερο να ολοκληρώνεται η δουλειά από εκείνον που την έχει ξεκινήσει.
- Αύξηση της προσωπικής επικοινωνίας ώστε να γίνεται η μετάδοση της «σιωπηρής γνώσης» ευκολότερα.
- Στη γραπτή επικοινωνία, εκτός των τυπικών και λεπτομερών τεκμηριώσεων, θα μπορούσαν να προστεθούν λιγότερο τυπικές τεκμηριώσεις που θα περιλαμβάνουν αυτά τα βασικά σημεία προσοχής. Π.χ. κάποια Hints ή Tips που πρέπει να έχει κανείς στο μυαλό του για τη συγκεκριμένη εφαρμογή.

### **6.6 Εναλλαγή εργασίας (Task switching ).**

Το SW Development απαιτεί βαθιά συγκέντρωση από τον προγραμματιστή/αναλυτή. Η εναλλαγή εργασιών εκτός του να αποσπά τον εργαζόμενο για κάποιο χρονικό διάστημα, μπορεί να οδηγήσει και σε υποδεέστερο ποιοτικά προϊόν. Όταν οι «εργάτες γνώσης» ασχολούνται παράλληλα με αρκετές εργασίες, σπαταλάνε πολύ χρόνο κατά την πνευματική μετακίνησή τους από τη μία δουλειά στην άλλη.

Προφανώς στο SW Development είναι απίθανο να μη γίνει τέτοια εναλλαγή, π.χ. λόγω έκτακτης ανάγκης ή σημαντικού προβλήματος. Μερικές ιδέες που θα μπορούσαν να βοηθήσουν είναι :



- Για ορισμένο χρόνο μέσα στην ημέρα να ασχολείται η ομάδα ανάπτυξης με τα θέματα που έχουν προκύψει από την προηγούμενη και στη συνέχεια να καθορίζεται το ημερήσιο πλάνο υλοποίησης.
- Κατηγοριοποίηση των αιτημάτων υποστήριξης και άμεση επίλυση μόνο των εξαιρετικά επείγοντων. Οι υπόλοιπες εκκρεμότητες θα πρέπει να δρομολογούνται και να προγραμματίζονται σε εβδομαδιαίο-μηνιαίο ορίζοντα ή σύμφωνα με τις ανάγκες.

### 6.7 Καθυστερήσεις-Αναμονές(delays)

Η αναμονή για τη διαθεσιμότητα ατόμων είναι βασικό σημείο σπατάλης στο SW. Οι προγραμματιστές λαμβάνουν σημαντικές αποφάσεις για την υλοποίηση σχεδόν κάθε 15 λεπτά και φυσικά αυτό δεν σημαίνει ότι όλες οι απαραίτητες πληροφορίες για αυτές τις αποφάσεις υπάρχουν σε κάποιο έντυπο τεκμηρίωσης ή προδιαγραφών. Μια προγραμματιστική απόφαση μπορεί να παρθεί εύκολα από τον προγραμματιστή εάν έχει απόλυτη κατανόηση του τι προσπαθεί να υλοποιήσει ή εάν υπάρχει κάποιος ο οποίος μπορεί να του δώσει την απάντηση άμεσα. Σε άλλη περίπτωση οι επιλογές που έχει είναι οι εξής : να σταματήσει την υλοποίηση και να ψάξει την πληροφορία που είναι απαραίτητη, να ασχοληθεί με κάποια άλλη εργασία, ή να αποφασίσει μόνος του βασιζόμενος στο ένστικτο, την εμπειρία και την τύχη. Προφανώς καμία από τις παραπάνω προσεγγίσεις δεν είναι καλή.

Οι «δεμένες» ομάδες που επικοινωνούν συχνά για το έργο που υλοποιούν και δίνουν άμεσες απαντήσεις, μειώνουν κατά πολύ τέτοιου είδους καθυστερήσεις και βεβαίως αυξάνουν την ποιότητα της ληφθείσας απόφασης. Άσχετα από τη φυσική τοποθεσία κάθε μέλους της ομάδας, είναι απαραίτητο να δίνεται η απαραίτητη πληροφορία τη στιγμή που ζητείται, όχι νωρίτερα (διότι πιθανά θα αλλάξει), ούτε και αργότερα (διότι πιθανά θα είναι άκαιρη).

Καθυστερήσεις δεν προκαλούνται μόνο από την αναμονή απαντήσεων, αλλά και από πολλές άλλες αιτίες. Επίσης κάθε οργανισμός μπορεί να αντιμετωπίζει διαφορετικού είδους καθυστερήσεις. Το σημαντικό είναι να τις αναγνωρίσει και να τις μειώσει. Μερικές ενδεικτικές πηγές καθυστέρησης στο SW Development είναι οι:

- Διαδικασίες εγκρίσεων projects
- Διαδικασίες εγκρίσεων αλλαγών

- Ανάθεση ατόμων
- Διαθεσιμότητα ατόμων

### 6.8 Ελαττωματικός κώδικας (Defects).

Ο κώδικας θα πρέπει να ελέγχεται ώστε να είναι mistake-proof, δηλ. να «προφυλάσσει» το χρήστη από λάθη που είναι πιθανό να κάνει. Επίσης να είναι ελεγμένος για απρόβλεπτα λάθη, «σκασίματα», υπερφόρτωση, κλπ.

Η ομάδα υλοποίησης οφείλει να θεσπίσει διαδικασίες ελέγχου, να βρει κανόνες που θα ακολουθούνται από όλους και να κάνει χρήση εργαλείων που θα τη βοηθήσουν στη διασφάλιση της ποιότητας και ορθότητας του κώδικα.

Η εύρεση των λαθών θα πρέπει να γίνεται νωρίς από τους προγραμματιστές/αναλυτές ώστε να μη δίνεται στους χρήστες προς έλεγχο κώδικας με ασταθή συμπεριφορά. Εάν κατά τα UAT(User Acceptance Tests) η εύρεση bugs από τους χρήστες είναι συχνό φαινόμενο, τότε πρέπει να ελεγχτεί και να αναδιοργανωθεί η διαδικασία υλοποίησης.

Στην κριτική ότι ο εξονυχιστικός έλεγχος στο στάδιο της κωδικοποίησης μπορεί να είναι χρονοβόρος, η απάντηση είναι πως η πρόληψη είναι πάντα η καλύτερη θεραπεία. Το φρόνιμο είναι να λειτουργεί ο οργανισμός προληπτικά (proactively) και όχι αντιδραστικά (reactively) και όπως λέει ο Shigeo Shingo ο έλεγχος για την **αποφυγή** ελαττωμάτων είναι απολύτως απαραίτητος σε κάθε διαδικασία, ενώ ο έλεγχος για την **εύρεση** των υπάρχοντων ελαττωμάτων είναι σπατάλη.

### 6.9 Υποαπασχόληση.

Είναι συχνό το φαινόμενο λόγω κακών διαδικασιών οι εργαζόμενοι στο SW να υποαπασχολούνται. Μερικές αιτίες είναι οι παρακάτω :

- επαναληψιμότητας βαρετών εργασιών (που θα μπορούσαν να αυτοματοποιηθούν)
- γραφειοκρατικές διαδικασίες εγκρίσεων που αποτρέπουν την καινοτομία και τη βελτίωση
- έλλειψη καθοδήγησης, στόχων, μετρητών απόδοσης ώστε να ενδυναμωθούν οι εργαζόμενοι και να ξέρουν προς τα που πρέπει να κινηθούν
- έλλειψη εκπαίδευσης

- έλλειψη επικοινωνίας μέσα στην ομάδα με αποτέλεσμα να μη μοιράζεται η γνώση και να μη γεννιούνται νέες συνεργατικές ιδέες.

### 6.10 Αποτύπωση του Value Stream.

Ο Taiichi Ohno συνοψίζει το Toyota Production System ως : «Αυτό που κάνουμε είναι να ελέγχουμε το χρονοδιάγραμμα(timeline) από τη στιγμή που ένας πελάτης δίνει μια παραγγελία έως τη στιγμή που λαμβάνουμε τα χρήματα. Και μειώνουμε αυτό το χρονοδιάγραμμα αφαιρώντας τις σπατάλες που δεν προσθέτουν αξία στον πελάτη.» Αυτό το χρονοδιάγραμμα μπορεί να απεικονιστεί ως ένα value stream, ένα διαγνωστικό εργαλείο που χρησιμοποιείται ευρέως στο lean και φυσικά μπορεί να χρησιμοποιηθεί και στο SW Development με τις κατάλληλες προσαρμογές.

Το value stream πάντα ξεκινάει και τελειώνει με τον πελάτη. Σε ένα περιβάλλον ανάπτυξης (κώδικα/προϊόντος) το ρολόι ξεκινάει τη στιγμή που ο πελάτης κάνει ένα αίτημα/δίνει μια παραγγελία (η έννοια της παραγγελίας μπορεί να διαφέρει από οργανισμό σε οργανισμό). Το ρολόι σταματάει όταν το πρόβλημα του πελάτη επιλύεται επιτυχώς (μεταφορά του SW στην παραγωγή κι επιβεβαίωση/λανσάρισμα του νέου προϊόντος). Το value stream καταγράφει τα βασικά γεγονότα που λαμβάνουν χώρα από τη στιγμή που το ρολόι αρχίζει να μετράει έως τη στιγμή που σταματάει.

Η αποτύπωση του value stream έχει αποδειχτεί ιδιαίτερα χρήσιμη στην ανάδειξη της σπατάλης, διότι οι καθυστερήσεις που παρουσιάζονται στη ροή είναι σχεδόν πάντα σημάδι σπατάλης.

Τα τέσσερα βασικά βήματα για τη δημιουργία ενός value stream είναι τα παρακάτω:

**Επιλογή value stream** προς διερεύνηση. Η συνιστώμενη πρακτική είναι να απεικονίζεται μια διαδικασία ή ένα είδος διαδικασιών και όχι ένα μεμονωμένο γεγονός. Π.χ. θα μπορούσε να απεικονιστεί το value stream της δημιουργίας μιας νέας λειτουργικότητας σε υπάρχουσα εφαρμογή με προτεραιότητα επείγοντος, ή το λανσάρισμα ενός νέου μεσαίου μεγέθους προϊόντος.

**Επιλογή έναρξης - λήξης.** Στην ανάπτυξη νέου προϊόντος συνήθως το ρολόι ξεκινάει όταν έχει παρθεί η απόφαση για την ανάπτυξή του. Όμως αυτή η χρονική στιγμή πιθανά να μη λαμβάνει υπόψιν σημαντικά βήματα που κάποια εταιρεία θα ήθελε να αξιολογήσει.

Άρα θα μπορούσε η έναρξη να είναι π.χ. όταν το τμήμα marketing διαπιστώνει την ανάγκη του νέου προϊόντος. Η λήξη αφορά την ικανοποίηση της ανάγκης του πελάτη ή το λανσάρισμα του νέου προϊόντος.

**Εύρεση/Ανάδειξη του value stream owner.** Το value stream μπορεί να σχεδιαστεί χωρίς value stream owner. Όμως η εμπειρία καταδεικνύει πως τα μεγαλύτερα προβλήματα ανακύπτουν σε οργανισμούς στους οποίους κανείς δεν είναι υπεύθυνος για το αίτημα του πελάτη/χρήστη. Το αίτημα μπορεί να περιμένει στην ουρά μέχρι να εγκριθεί, να περιμένει την μετακίνησή του από τμήμα σε τμήμα κι από λειτουργία σε λειτουργία, ή να περιμένει στην ουρά για μεταφορά στην παραγωγή. Μόνο αν υπάρχει ένας υπεύθυνος value stream owner που θα «τρέξει» το αίτημα του πελάτη/χρήστη, μπορούν να αντιμετωπιστούν οι παραπάνω πηγές σπατάλης.

### 6.11 Διατήρηση της απλότητας.

Η αποτύπωση του value stream είναι ένα διαγνωστικό εργαλείο εύρεσης της σπατάλης. Δεν προσθέτει από μόνη της κάποια αξία. Δε χρειάζεται ιδιαίτερη πολυπλοκότητα ούτε τεράστια λεπτομέρεια. Ούτε καν κάποιο σχεδιαστικό εργαλείο. Αρκούν ένα χαρτί κι ένα μολύβι.

Η σημασία της αποτύπωσης αυτής έγκειται στο ότι βοηθάει να δούμε μέσα από τα μάτια του πελάτη/χρήστη τη δική μας διαδικασία ανάπτυξης κώδικα ώστε να τη βελτιώσουμε.

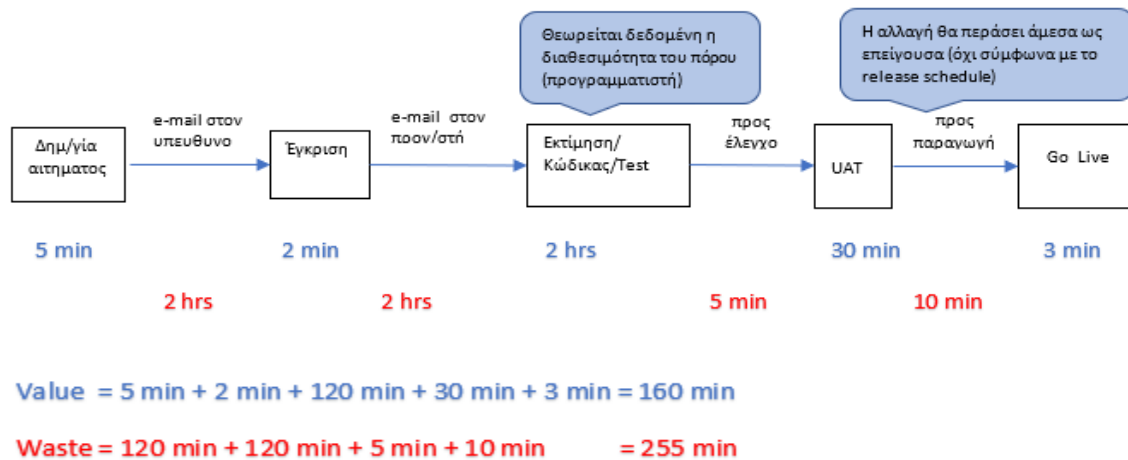
Ο σκοπός της αποτύπωσης είναι :

- να βρεθεί ο πραγματικός χρόνος που παρέρχεται από τη στιγμή που ο πελάτης/χρήστης δημιουργεί ένα αίτημα έως τη στιγμή που αυτό μεταφέρεται στην παραγωγή (πλήρως ελεγμένο κατά τα προαναφερόμενα) και
- να υπολογιστεί η **αποδοτικότητα** αυτού του value stream, δηλ. το ποσοστό του χρόνου που παράγει αξία σε σχέση με το συνολικό χρόνο (process cycle efficiency). (Michael George and Stephen Wilson, 2004).

Κατά το σχεδιασμό του Value stream έχει σημασία να μη χάσουμε το στόχο, ο οποίος είναι η αναγνώριση της σπατάλης και η εξάλειψή της.

Στη συνέχεια παραθέτουμε ένα παράδειγμα για πληρέστερη κατανόηση. Το παράδειγμα αφορά μια επείγουσα μικρή ενίσχυση σε κώδικα η οποία πρέπει να περάσει άμεσα στην

παραγωγή.



Σχήμα 11. Value stream map

Ο χρήστης δημιουργεί το σχετικό αίτημα το οποίο παραλαμβάνεται και εγκρίνεται από τον αρμόδιο υπεύθυνο σε 2 ώρες κατά μέσο όρο. Χρειάζονται γύρω στις 2 ώρες για την ολοκλήρωση και τον έλεγχο της αλλαγής από τον προγραμματιστή, θεωρώντας βέβαια δεδομένη τη διαθεσιμότητά του. Επίσης η αλλαγή θα περάσει άμεσα στην παραγωγή και όχι βάσει του αυτόματου scheduling, λόγω του επείγοντος χαρακτήρα.

Ο συνολικός χρόνος σε αυτήν την περίπτωση ήταν  $160 + 255 = 415$  min.

Η απόδοση του συγκεκριμένου value stream είναι ~ 39%. Προφανώς υπάρχει χώρος για βελτίωση αφού οι εγκρίσεις/αναθέσεις διαρκούν περίπου 4 ώρες.

Αντίστοιχες απεικονίσεις value stream μπορούν να γίνουν και για άλλες διαδικασίες με απώτερο στόχο την εύρεση της σπατάλης, που μπορεί να προκαλείται λόγω ουράς ή ατέρμωνων κύκλων π.χ. test και διορθώσεις bugs. Ο στόχος είναι να γίνουν εκείνες οι διορθωτικές κινήσεις που θα βελτιώσουν την απόδοση στοχεύοντας πάντα στην αύξηση της αξίας για τον πελάτη.

## **ΚΕΦΑΛΑΙΟ 7. ΟΙ ΑΝΘΡΩΠΟΙ ΣΤΗ LEAN ΦΙΛΟΣΟΦΙΑ**

### **7.1 Εισαγωγή (Lean system of management).**

Το Lean ως σύστημα διοίκησης έχει σκοπό τη δημιουργία σκεπτόμενων και αφοσιωμένων στο στόχο ανθρώπων, σε όλα τα επίπεδα της ιεραρχίας και ειδικότερα στην πρώτη γραμμή. Εάν κάποια εταιρεία υλοποιήσει όλες τις αρχές του Lean εκτός του σεβασμού στον άνθρωπο, θα αποκομίσει πολύ λιγότερα από τα δυνητικά οφέλη που θα μπορούσε να έχει. Εάν εφαρμόσει όμως αυτήν τη μία και μοναδική αρχή, είναι βέβαιο ότι οι άνθρωποι της εταιρείας θα ανακαλύψουν και θα υλοποιήσουν τις υπόλοιπες.

Όπως έχει αναφερθεί στο κεφάλαιο της Ιστορικής Αναδρομής, ο Edward Deming βοήθησε τις Ιαπωνικές εταιρείες να ανακάμψουν μετά το 2<sup>ο</sup> παγκόσμιο πόλεμο. Οι Ιαπωνικές εταιρείες επιζητούσαν να αλλάξει η διαδεδομένη αντίληψη της εποχής ότι τα ιαπωνικά προϊόντα ήταν υποδεέστερης ποιότητας. Ο Deming βοήθησε τους Ιάπωνες με συμβουλές οι οποίες στηρίζονταν στη θεωρία του πως η παραγωγή είναι ένα «σύστημα» που περιλαμβάνει τον προμηθευτή και τον πελάτη, κι ότι οι ασκούντες την ηγεσία θα πρέπει να εστιάζουν στη συνεχή βελτίωση του συνολικού συστήματος. Ένας τρόπος που τους δίδαξε για τη βελτίωση αυτή, ήταν η χρήση στατιστικού ελέγχου ποιότητας ώστε να «χτίζεται» η ποιότητα στα παραγόμενα προϊόντα και όχι να ελέγχεται η ποιότητά τους εκ των υστέρων.

Στη συνέχεια θα γίνει αναφορά σε βασικές αρχές που αφορούν τους ανθρώπους-εργαζόμενους ως μέλη του οργανισμού (Mary and Tom Poppendieck, 2007). Οι lean αυτές αρχές αφορούν γενικότερα τους εργαζόμενους σε μια εταιρεία και όχι μόνο το SW Development, αλλά είναι πολύ σημαντικές και αξίζει να τις εξετάσουμε.

### **7.2 Σύστημα Deming.**

Ο Deming(2000) καθιέρωσε ένα σύστημα γνώσης το οποίο είχε τέσσερις βασικούς πυλώνες:

#### **Εκτίμηση απέναντι στο σύστημα. (Appreciation for a system)**

Θεωρούσε ότι οι συνέργειες ανάμεσα στα επιμέρους τμήματα ενός συστήματος είναι το σημείο κλειδί για τη συνολική επιτυχία του συστήματος. Εφόσον λοιπόν η παραγωγή αντιμετωπιζόταν ως σύστημα, ήταν πολύ σημαντική η διοίκηση της σχέσης ανάμεσα στα

επιμέρους τμήματα (ή της σχέσης ανάμεσα στα τμήματα μιας εταιρείας), δηλ. τη σχέση ανάμεσα σε προμηθευτές, παραγωγούς και πελάτες. Πίστευε ότι το σύστημα έπρεπε να αντιμετωπίζεται ως «όλον» κι ότι βελτίωση των επιμέρους τμημάτων δεν ήταν προς όφελος του συνολικού συστήματος.

### **Κατανόηση των παρεκκλίσεων. (Knowledge about variation)**

Ο Deming δε συμφωνούσε όταν οι εργάτες κατηγορούνταν για λάθη και προβλήματα τα οποία στην πραγματικότητα ήταν προβλήματα του συστήματος στο οποίο δούλευαν. Στις διαλέξεις που έδινε, προσπαθούσε να εξηγήσει στους διοικούντες ότι οι βασικότερες και περισσότερες αιτίες για την κακή ποιότητα και τη μειωμένη παραγωγικότητα οφείλονταν σε συστημικά προβλήματα και όχι σε ατομικά λάθη/παραλείψεις των εργατών. Οι προθεσμίες (deadlines) και η αυστηρή διοίκηση δεν αντιμετώπιζαν το συστημικό πρόβλημα. Αυτό που έπρεπε να γίνει σύμφωνα με τον Deming, ήταν η ηγεσία να αλλάξει τον τρόπο λειτουργίας του συστήματος.

### **Θεωρία της Γνώσης. (Theory of Knowledge)**

Ο Deming ως φυσικός, ασπαζόταν την επιστημονική μέθοδο κατανόησης ενός συστήματος καθώς και της λήψης αποφάσεων. Τα βήματα της επιστημονικής μεθόδου (υπόθεση-πείραμα-μάθηση-ενσωμάτωση της μάθησης), αντικατοπτρίζουν το γνωστό κύκλο του Deming (Plan-Do-Check-Act. PDCA). (Σημειώνεται ότι ο Deming τον αποκαλούσε κύκλο του Shewhart, διότι τον είχε εφεύρει ο Walter Shewhart τη δεκαετία του 1930).

### **Ψυχολογία. (Psychology)**

Ο Deming προειδοποιούσε τους managers ότι τα νούμερα δε λένε όλη την αλήθεια, κι ότι όσον αφορά στους ανθρώπους, αυτά που μετράνε και κάνουν τη διαφορά είναι οι δεξιότητες, η υπερηφάνεια, η αυτοπεποίθηση και η συνεργασία.

Βάσει των παραπάνω πυλώνων, ο Deming έθεσε δεκατέσσερις βασικές αρχές όσον αφορά το Management μιας εταιρείας :

- i. Πρέπει να θέτει μακροπρόθεσμους στόχους και να μην εστιάζει σε βραχυπρόθεσμα κέρδη. Ο σκοπός είναι να έχει μακροβιότητα ή εταιρεία και να παρέχει θέσεις εργασίας.

- ii. Ο κόσμος αλλάζει και οι managers πρέπει να προσαρμόζονται σε νέους τρόπους σκέψης. Καθυστερήσεις, λάθη, ελαττωματικά προϊόντα, χαμηλής ποιότητας υπηρεσίες, δεν είναι αποδεκτά.
- iii. Πρέπει να σταματήσει ο εκ των υστέρων έλεγχος ποιότητας για την εύρεση προβλημάτων, αλλά να χτίζεται η ποιότητα μέσα στα προϊόντα κατά τη δημιουργία τους.
- iv. Η επιλογή των προμηθευτών δεν πρέπει να γίνεται μόνο σε οικονομική βάση. Η μείωση του συνολικού κόστους μπορεί να επέλθει με τη δημιουργία μακροχρόνιων σχέσεων με τους προμηθευτές, οι οποίες βασίζονται στην πίστη και την εμπιστοσύνη.
- v. Η προσπάθεια βελτίωσης του συστήματος παραγωγής και υπηρεσιών πρέπει να είναι συνεχής. Κάθε λειτουργία του συστήματος πρέπει να βελτιώνεται με στόχο τη μείωση της σπατάλης και την αύξηση της ποιότητας.
- vi. Η εκπαίδευση σε πρώτο πλάνο. Οι managers οφείλουν να γνωρίζουν πώς εκτελείται η εργασία την οποία επιβλέπουν και να μπορούν να εκπαιδεύσουν τους εργαζόμενους. Αλλά και οι managers πρέπει να εκπαιδεύονται ώστε να αντιλαμβάνονται και να κατανοούν το σύστημα παραγωγής.
- vii. Ενίσχυση της ηγεσίας. Η δουλειά των managers είναι να βοηθάνε τους ανθρώπους τους να κάνουν καλύτερη δουλειά και να αποβάλλουν στοιχεία του συστήματος (συμπεριφορές-κουλτούρα κλπ.), που τους εμποδίζουν να κάνουν τη δουλειά τους με υπερηφάνεια.
- viii. Χτίσιμο σχέσεων μη φόβου. Βασική ανάγκη του ανθρώπου είναι να νιώθει ασφαλής στον εργασιακό του χώρο, ώστε να αποδίδει καλύτερα. Για να γίνει αυτό, το διοικητικό περιβάλλον πρέπει να είναι βοηθητικό και όχι απειλητικό.
- ix. Κατάργηση των φραγμών ανάμεσα στα τμήματα. Είναι απαραίτητη η δημιουργία διαλειτουργικών ομάδων, ώστε το κάθε μέλος να αντιλαμβάνεται την οπτική των άλλων μελών.



- x. Προσοχή στη χρήση προτροπών και στόχων. Το σύστημα παράγει τα ελαττωματικά προϊόντα και τη χαμηλή παραγωγικότητα, όχι οι εργάτες(προγραμματιστές). Με τις προτροπές δεν αλλάζει το σύστημα. Αυτή είναι δουλειά της διοίκησης.
- xi. Εξάλειψη των αριθμητικών ποσοστώσεων/στόχων. (και των αυθαίρετων προθεσμιών για τους προγραμματιστές). Αυτό είναι διοίκηση μέσω εκφοβισμού. Πρέπει να αντικατασταθεί με διοίκηση ηγεσίας (leadership).
- xii. Κατάργηση συνηθειών/συμπεριφορών που στερούν την αξιοπρέπεια του εργαζόμενου και την υπερηφάνεια για τη δουλειά του. Οι ωρομίσθιοι υπάλληλοι δεν πρέπει να θεωρούνται εμπόρευμα και για τους μισθωτούς πρέπει να καταργηθεί η ετήσια βαθμολόγηση.
- xiii. Η εκπαίδευση και η αυτοβελτίωση πρέπει να ενισχύονται και να ενθαρρύνονται. Ένα καλά εκπαιδευμένο εργατικό δυναμικό τόσο στα χαμηλότερα επίπεδα ιεραρχίας όσο και στα υψηλότερα, είναι εγγύηση για το μέλλον.
- xiv. Ο μετασχηματισμός του οργανισμού (σε lean) πρέπει να γίνει με δράσεις και πράξεις. Η κορυφή της διοίκησης πρέπει να είναι οδηγός και ηγέτης σε αυτήν την προσπάθεια και όχι απλός υποστηρικτής.

### **7.3 Λόγοι αποτυχίας των προγραμμάτων μετασχηματισμού.**

Πολλές επιχειρήσεις έχουν κατά καιρούς προσπαθήσει να υλοποιήσουν προγράμματα μετασχηματισμού και βελτίωσής τους, βασισμένα σε διάφορες αρχές όπως του Deming, του Lean ή του Six Sigma. Αυτές οι προσπάθειες ξεκινούσαν πάντα με τις καλύτερες προθέσεις και όντως έφερναν αποτέλεσμα τουλάχιστον τον πρώτο καιρό εφαρμογής τους. Στην πορεία όμως έχαναν την αρχική τους δυναμική και έπαιρναν πιο γραφειοκρατικό χαρακτήρα. Τα αποτελέσματα τέτοιων προγραμμάτων κυμαίνονταν από εξαιρετικά σε κάποιες εταιρείες έως μέτρια ή απογοητευτικά σε άλλες.

Σύμφωνα με τον Gary Hamel(2006), οι μεγαλύτερες βελτιώσεις στις εταιρείες προέρχονται από διοικητικές καινοτομίες(management innovation), που ξεφεύγουν από τις παραδοσιακές διοικητικές (management) αρχές, διαδικασίες ή οργανωτικές δομές.

Τέτοιου είδους διοικητικές καινοτομίες προσδίδουν τεράστιο ανταγωνιστικό πλεονέκτημα στον οργανισμό, το οποίο δεν μπορεί εύκολα να αντιγραφεί από τον ανταγωνισμό.

Αυτό βέβαια δεν αποτρέπει τις εταιρείες από το να αντιγράφουν μεθόδους και πρακτικές επιτυχημένων ανταγωνιστών. Προσπαθούν λοιπόν να μιμηθούν την επιτυχημένη διοικητική καινοτομία, σύμφωνα όμως με τη δική τους αντίληψη. Σχεδόν πάντα, παραβλέπεται από την εταιρεία-μιμητή η θεμελιώδης αρχή της εκάστοτε πρωτότυπης καινοτομίας. Επειδή ο «ορθός» τρόπος διοίκησης είναι βαθιά ριζωμένος στη νοοτροπία των περισσότερων executives, δεν μπορούν να αφομοιώσουν τέτοιου είδους αλλαγές. Όσο πιο ριζοσπαστική είναι η διοικητική καινοτομία, τόσο πιο δύσκολο είναι να αντιγραφεί ώστε να έχει αποτέλεσμα στην εταιρεία-αντιγραφέα.

Ουσιαστικά το κεντρικό σημείο, η κεντρική ιδέα για την οποία υλοποιήθηκε η διοικητική καινοτομία, είναι άγνωστη και αόρατη στην εταιρεία-μιμητή. Για παράδειγμα πολλές εταιρείες έχουν προσπαθήσει να εφαρμόσουν τις Lean αρχές, αλλά απέτυχαν διότι έχαναν την κεντρική ιδέα. Όπως αναφέρει ο Hamel, η πραγματική καινοτομία της Toyota είναι η ικανότητά της να αξιοποιεί τη διάνοια «απλών» υπαλλήλων. Για να είναι οι lean μετασχηματισμοί επιτυχημένοι, πρέπει πρώτα και πάνω απ' όλα να βασίζονται στο σεβασμό για κάθε άνθρωπο στην εταιρεία, ειδικά για τους «απλούς» υπαλλήλους που δημιουργούν το προϊόν ή τον κώδικα.

Η έννοια του σεβασμού δεν είναι κάτι απλό και εύκολο. Πρέπει να συνάδουν τα λόγια με τις πράξεις. Π.χ. αν η διοίκηση έχει ως σημαία την ποιότητα αλλά ταυτόχρονα «κυνηγεί» μόνο προθεσμίες, αν προτρέπει τους υπαλλήλους σε ομαδική συνεργασία αλλά τους αξιολογεί ατομικά, αν πρεσβεύει την εμπιστοσύνη προς τους προμηθευτές αλλά συνάπτει μόνο συμβόλαια σταθερής τιμής (fixed-price), τότε χάνει τη βασική αρχή του lean, του σεβασμού προς τον άνθρωπο.

#### **7.4 Η έννοια της ομάδας.**

Στους περισσότερους οργανισμούς η έννοια της ομαδικότητας και της συνεργασιμότητας βρίσκεται ψηλά στην κατάταξη των αρχών που, θεωρητικά τουλάχιστον, τους διέπουν.

Όσον αφορά το SW ειδικότερα, η δημιουργία του είναι πάντα αποτέλεσμα ομαδικής δουλειάς. Απαιτεί συνεχή επίλυση προβλημάτων και λήψη δύσκολων αποφάσεων που επηρεάζουν και άλλους μέσα ή και εκτός ομάδας και υλοποιούνται με συνεργασία.

Οι ομάδες έχουν ανάγκη την πρόκληση, τον κοινό στόχο και την αμοιβαία δέσμευση να δουλέψουν για την εκπλήρωση του στόχου. Τα μέλη της ομάδας εξαρτώνται το ένα από το άλλο και αλληλοβοηθούνται. Οι εταιρείες θα πρέπει να στοχεύουν στη δημιουργία περιβάλλοντος στο οποίο οι ομάδες θα αναπτύσσονται και θα δημιουργούν αξία.

Όπως προείπαμε κάτι τέτοιο δεν είναι εύκολο. Τα συστήματα αξιολόγησης που προωθούν την ατομικότητα, συμμετοχή μελών σε παραπάνω από δύο ομάδες ή υπεύθυνοι που δεν μπορούν να υποστηρίξουν και να ελέγξουν την ομαδική εργασία, είναι μερικά παραδείγματα προς αποφυγή στο δρόμο προς την ομαδικότητα.

## 7.5 Εξειδίκευση.

Η Toyota πιστεύει ότι η σταθερή και συνεχής τελειότητα στην παραγωγή αρχίζει με την υψηλή τεχνική επάρκεια όλων των μηχανικών.(James Morgan-Jeffrey Liker, 2006). Πιστεύει επίσης ότι η επάρκεια αυτού του επιπέδου δε μαθαίνεται στις σχολές, αλλά αναπτύσσεται με την πάροδο του χρόνου, με την εμπειρία και την καθοδήγηση. Ο νέος μηχανικός στην Toyota περνάει αρκετά χρόνια σε διάφορες θέσεις έως ότου αποκτήσει τις απαραίτητες δεξιότητες για να θεωρείται first-rate μηχανικός.

Στην Toyota η ανάπτυξη των ικανοτήτων των υπαλλήλων είναι θεμελιώδες στοιχείο της δουλειάς του manager. Όλοι οι managers αντιμετωπίζουν την απόδοση των ομάδων τους ως άμεση απόρροια της δικής τους ικανότητας.

Στο χώρο του SW τα εξαιρετικά λογισμικά δημιουργούνται από ανθρώπους με μεγάλη τεχνική γνώση. Στο lean αναγνωρίζεται αυτή η σημαντικότητα και πρέπει να είναι εταιρικός στόχος η πρόσληψη εξειδικευμένων επαγγελματιών καθώς και η συνεχής τεχνική τους ανάπτυξη.

Θα πρέπει να υπάρχουν οι δομές και οι διαδικασίες ώστε ο νεοεισερχόμενος developer (αλλά και οποιοσδήποτε νέος υπάλληλος), να ενσωματώνεται ομαλά στην ομάδα και να μαθαίνει από τους παλαιότερους τις διαδικασίες, την κουλτούρα, τα μυστικά της δουλειάς κλπ. Δε θα πρέπει να υπάρχουν φραγμοί που θα τον αποκλείουν από την ομάδα επειδή ακόμα δεν έχει τις απαραίτητες δεξιότητες. Είναι σίγουρο ότι με την κατάλληλη καθοδήγηση θα ενσωματωθεί και θα αποδώσει πολύ γρηγορότερα. Επίσης και οι παλαιότεροι έχουν να λαμβάνουν από τους νέους συναδέλφους, εκτός από τις πιθανές νέες τεχνολογικές γνώσεις που διαθέτουν, την καινούργια, φρέσκια αύρα που συνήθως

φέρνουν μαζί τους οι νεότεροι. Οι ομάδες εργασίας θα πρέπει να είναι μικτές και να περιλαμβάνουν και έμπειρους συναδέλφους αλλά και νεότερους. Φυσικά η όποια αναγνώριση για το αποτέλεσμα αφορά όλη την ομάδα.

## 7.6 Ηγεσία.

Ο ρόλος της ηγεσίας είναι πολύ σημαντικός στο να φέρει μία ομάδα το επιθυμητό αποτέλεσμα. Επαγγελματική ομάδα χωρίς ηγέτη είναι σαν ομάδα κωπηλασίας χωρίς πηδαλιούχο. Βεβαίως οι πολύ καλοί αθλητές της ομάδας θα κωπηλατήσουν γρήγορα, αλλά δεν θα μπορέσουν να συγχρονιστούν ή να πάρουν τις κατάλληλες αποφάσεις ώστε να κερδίσουν τον αγώνα. Η Toyota λειτουργεί με τη βασική αρχή ότι η ομαδική δουλειά είναι απολύτως ουσιαστική, αλλά πάντα πρέπει να υπάρχει ένας υπεύθυνος για την ομάδα.

Το ίδιο ισχύει και στο SW Development. Ο υπεύθυνος έχει ως στόχο να κάνει ξεκάθαρους τους στόχους της ομάδας, να την κινητοποιήσει και να την υποστηρίξει. Πιο αναλυτικά :

- Να κάνει σαφή το σκοπό που πρέπει να επιτελέσει η ομάδα και ταυτόχρονα να ανυψώνει το ηθικό της. Τα μέλη που δεσμεύονται για έναν κοινό στόχο και ταυτόχρονα αισθάνονται ενθουσιασμό, θα συνεργαστούν με αφοσίωση και ζήλο για την επίτευξή του.
- Να είναι σίγουρος ότι ο στόχος μπορεί να επιτευχθεί και ότι η ομάδα έχει την ικανότητα να ανταπεξέλθει. Εάν η ομάδα χρειάζεται πρόσβαση σε πληροφορίες ή άλλους πόρους, ο υπεύθυνος έχει καθήκον να φροντίζει για την προσβασιμότητα τους. Επίσης έχει το καθήκον της ενδυνάμωσης της ομάδας, όπου χωλαίνει.
- Να επιτρέπει την πρόσβαση προς τους πελάτες. Είναι πολύ σημαντικό για τους προγραμματιστές να έχουν άμεση εικόνα του πώς θα βοηθηθούν οι πελάτες/χρήστες από τη δουλειά τους.
- Να επιτρέπει στην ομάδα τον προγραμματισμό της δουλειάς. Η ομάδα θα δεσμευτεί για την παράδοση συγκεκριμένων λειτουργιών, εφόσον έχει επικοινωνήσει με τους πελάτες και έχει ελέγξει το πρόγραμμά της.
- Να κάνει τις κατάλληλες ενέργειες που υποδεικνύει η ομάδα. Μια σοβαρή και κινητοποιημένη ομάδα δε χρειάζεται κάποιον να της λέει τι θα κάνει. Μπορεί όμως να δώσει τις κατάλληλες υποδείξεις, τις οποίες ο υπεύθυνος θα πρέπει να

αξιολογήσει κατάλληλα και να υλοποιήσει ή όχι. Ένα τέτοιο παράδειγμα είναι π.χ. η ανάγκη ορθών δεδομένων στο test περιβάλλον ώστε οι έλεγχοι να είναι ακριβείς και τα αποτελέσματα όσο γίνεται πιο ασφαλή. Είναι ευθύνη του manager να ασχοληθεί με την ενημέρωση του test περιβάλλοντος, εφόσον έλαβε το μήνυμα από την ομάδα. Επίσης είναι δουλειά του manager να «βγαίνει μπροστά», να υποστηρίζει και να προστατεύει την ομάδα σε πιθανές διαμάχες με τους πελάτες/χρήστες π.χ. για τους χρόνους ή το παραγόμενο έργο.

## 7.7 Αξιολόγηση.

Θα αναφερθούμε στο θέμα της αξιολόγησης επειδή είναι ένα πολύ κρίσιμο σημείο για κάθε εργασιακό χώρο αλλά ειδικότερα για το SW Development.

Συνήθως η προτροπή του Deming για την κατάργηση της ετήσιας αξιολόγησης των μισθωτών αγνοείται από τις επιχειρήσεις. Ο Deming θεωρούσε ότι αυτή η ατομική αξιολόγηση υπονομεύει την ομαδική εργασία και προκαλεί ανταγωνισμό αντί για συνεργασία και φυσικά καταστρέφει το εργασιακό κλίμα ανάμεσα στους συναδέλφους (προγραμματιστές και μη).

Σε μια εταιρεία που αναπτύσσει τους υπαλλήλους της ακολουθώντας τη lean φιλοσοφία, οι ετήσιες αξιολογήσεις, θα πρέπει να αντικατασταθούν με συχνότερες συναντήσεις στις οποίες θα ανιχνεύονται πιθανά προβλήματα και οι λύσεις τους, ώστε να αυξηθεί η ατομική απόδοση σε συνάρτηση πάντα με την ομαδική. Σε αυτές τις συναντήσεις το θέμα της βελτίωσης δεν θα αφορά μόνο τον εργαζόμενο αλλά και τον manager. Θα πρέπει κι εκείνος να αναρωτηθεί και να ακούσει τι κάνει λάθος και πώς αυτό μπορεί να βελτιωθεί.

Ο στόχος όλων των εργαζομένων, διοίκησης και υπαλλήλων, πρέπει να είναι η βελτίωση του συνολικού συστήματος.

## **ΚΕΦΑΛΑΙΟ 8. ΟΔΗΓΟΣ LEAN ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΥ ΣΤΟ IT ΚΑΙ ΣΤΟ SW**

Στο κεφάλαιο αυτό ολοκληρώνοντας την ανάλυση της Lean Φιλοσοφίας στο Software Development, θα παρουσιάσουμε συνοπτικά τα βασικά βήματα που πρέπει να κάνει ένας SW development οργανισμός ο οποίος ενστερνίζεται τη Lean φιλοσοφία και τα οποία έχουν αναλυθεί στα προηγούμενα κεφάλαια.

### **8.1 Πυλώνες Lean μετασχηματισμού.**

Αρχικά θα αναφερθούμε στους βασικούς πυλώνες της επιτυχίας ενός Lean μετασχηματισμού στο SW development, αλλά και στο IT γενικότερα, που σύμφωνα με τους Mary και Tom Poppendieck (2007) είναι οι εξής:

- η εκπαίδευση των υπευθύνων (team leaders – supervisors)
- ο τρόπος σκέψης (on-the-job thinking )
- η μέτρηση ορισμένων βασικών δεικτών.

### **8.2 Εκπαίδευση υπευθύνων.**

Οι υπεύθυνοι πρέπει να εκπαιδεύονται ώστε να αποκτούν τις ικανότητες και τις δεξιότητες εκείνες με τις οποίες θα ενισχύουν τους υφισταμένους τους μέσω coaching. Πιο συγκεκριμένα οι υπεύθυνοι εκπαιδεύονται στα παρακάτω:

- στον τρόπο μεταλαμπάδευσης της γνώσης προς τους άλλους. Είναι συχνό φαινόμενο κάποιος να είναι πολύ αποδοτικός και αποτελεσματικός στη δουλειά του, αλλά να μην μπορεί να διδάξει και να μεταδώσει σε άλλους αυτό που ο ίδιος γνωρίζει πολύ καλά.
- να βοηθάνε τους υφισταμένους στο συντονισμό της δουλειάς.
- να υποστηρίζουν και ταυτόχρονα να επικροτούν την αναζήτηση καλύτερων μεθόδων / διαδικασιών από τους υφισταμένους.
- να επιλύουν δίκαια, ήρεμα και με σεβασμό προβλήματα και εντάσεις που αναπόφευκτα προκαλούνται στις ανθρώπινες σχέσεις.

### 8.3 Ο τρόπος σκέψης.

Σύμφωνα με τον Minoura Teruyuki, υψηλόβαθμο στέλεχος της Toyota, το T στο TPS σημαίνει “Thinking”, εκτός από Toyota. Με τον όρο thinking εννοεί τον τρόπο με τον οποίο η εταιρεία αναπτύσσει τους ανθρώπους της. Χαρακτηριστικά αναφέρει : «Κάτω από ένα “push” σύστημα, είναι μικρή η πιθανότητα για τους εργαζόμενους να αποκτήσουν σοφία(wisdom) διότι απλά παράγουν σύμφωνα με τις οδηγίες που έχουν λάβει. Αντίθετα ένα “pull” σύστημα επιζητεί(asks) από τον εργαζόμενο να χρησιμοποιήσει το μυαλό του ώστε να βρει μια διαδικασία όπου ο ίδιος αποφασίζει πώς πρέπει να γίνει και πόσο γρήγορα.».

Όταν οι εταιρείες εφαρμόζουν προγράμματα βελτίωσης διαδικασιών συνήθως δίνεται έμφαση στο documentation και στην εύρεση της βέλτιστης διαδικασίας και όχι στο να βοηθηθούν και ενθαρρυνθούν οι υπάλληλοι στο να κάνουν οι ίδιοι σκέψεις βελτίωσης. Βεβαίως συχνά οι εταιρείες διαθέτουν «θυρίδα προτάσεων», στην οποία μπορεί ο εργαζόμενος να εκφράσει ιδέες για βελτιώσεις. Επίσης συχνά όμως, αυτή η θυρίδα απλά μεγαλώνει ως λίστα και κρατάει την ιστορικότητα των προτάσεων, χωρίς ουσιαστικό αποτέλεσμα, πράγμα που ισχύει και στο SW development και στο IT.

Το πρόβλημα είναι πως οι βελτιωτικές προτάσεις κατατίθενται σε κάποιον άλλον για αξιολόγηση και υλοποίηση. Θα έπρεπε η ομάδα που κάνει την πρόταση να την υλοποιεί, έχοντας προφανώς την κατάλληλη αρμοδιότητα και ελευθερία. Όταν οι ιδέες υλοποιούνται από άλλους, ο αρχικός εμπνευστής χάνει το ζήλο του και πιθανά χάνεται και η σιωπηρή γνώση που ο ίδιος κατέχει για το πρόβλημα.

### 8.4 Μετρήσεις.

Ειδικά στο SW Development αλλά και στο IT, η εύρεση αξιόπιστων μετρητών απόδοσης (KPI) έχει αποδειχτεί ιδιαίτερα δύσκολη. Έτσι έχουν δημιουργηθεί αρκετοί μετρητές που μετράνε τα αποτελέσματα των επιμέρους τμημάτων των διαδικασιών και όχι το σύνολο. Αυτό όπως πλέον γνωρίζουμε από τη lean φιλοσοφία, είναι λάθος. Οι βελτιώσεις των τμημάτων δεν έχουν την ίδια απόδοση στο σύνολο.

Η προσπάθεια βελτίωσης των «λάθος» μετρητών, προκαλεί γενικότερο πρόβλημα ακόμα και στην κουλτούρα της εταιρείας. Π.χ. οι project managers συχνά αξιολογούνται βάσει της Earned Value. Όμως η EV έχει να κάνει με τον έλεγχο του πλάνου του project. Δεν ασχολείται με το αν το συγκεκριμένο πλάνο είναι και το καλύτερο για τα επιθυμητά

αποτελέσματα. Έτσι πολύ συχνά παραδίδεται το software/προϊόν έγκαιρα και χωρίς να ξεφεύγει από τον προϋπολογισμό του, παρόλα αυτά όμως οι πελάτες/χρήστες είναι δυσαρεστημένοι, διότι το προϊόν που παρέλαβαν δεν καλύπτει τις ανάγκες τους.

Για τους παραπάνω λόγους, προτείνονται ως KPIs : ο χρόνος κύκλου, τα οικονομικά αποτελέσματα και η ικανοποίηση του πελάτη. Παρακάτω θα δούμε την εφαρμογή τους στο SW Development.

Ο **χρόνος κύκλου** είναι ίσως η πιο σημαντική μέτρηση στην προσπάθεια βελτίωσης ενός οργανισμού. Αφορά το χρόνο που απαιτείται από την αίτηση του πελάτη έως την πλήρη ικανοποίηση του αιτήματος. Ο βασικός στόχος είναι η επίτευξη ενός αξιόπιστου, επαναλαμβανόμενου χρόνου κύκλου για κάθε τύπο αιτήματος (π.χ. χαμηλής/ μέτριας/ υψηλής προτεραιότητας), και στη συνέχεια αυτός ο χρόνος να μειώνεται.

Αυτή η σχετικά απλή μέτρηση, αναδεικνύει τα προβλήματα πάσης φύσεως όπως: τεχνικές αδυναμίες και άγνοια, έλλειψη προσωπικού, κακός κώδικας που δεν μπορεί εύκολα να αλλάξει, πολυπλοκότητα, κλπ.

Η προσπάθεια βελτίωσης του χρόνου κύκλου, απαιτεί τη συνεργασία όλων των εμπλεκόμενων και τη βελτίωση της ποιότητας του κώδικα καθώς και την απλούστευση των διαδικασιών.

Τα **οικονομικά αποτελέσματα** αφορούν στο κέρδος που θα αποκομίσει ο οργανισμός από την υλοποίηση του εκάστοτε project (είτε αυτό αφορά ένα πρόγραμμα είτε αυτοματοποίηση διαδικασιών κλπ.). Αυτά τα οικονομικά αποτελέσματα θα πρέπει να επικοινωνούνται στην ομάδα ανάπτυξης, ώστε να γίνεται κοινωνός της ευρύτερης προσπάθειας της εταιρείας. Επίσης θα πρέπει να ελέγχονται και να επικαιροποιούνται μετά την υλοποίηση, για πληρέστερο έλεγχο της απόδοσης του κάθε project. Δηλ. τελικά το project πέτυχε τους οικονομικούς του στόχους; και εάν όχι, γιατί ; Ο σκοπός εδώ είναι όχι να επιρριφθούν ευθύνες και ποινές, αλλά να εντοπιστούν τα λάθη και οι παραλείψεις, να γίνουν «μαθήματα» (lessons learnt), ώστε οι επόμενες προσπάθειες να είναι πιο επιτυχημένες.

Η **ικανοποίηση του πελάτη** είναι φυσικά αυτό που θεωρητικά επιζητούν όλοι οι οργανισμοί. Είναι αυτονόητο ότι οι βασικές απαιτήσεις οφείλουν να καλύπτονται. Ειδικά στο lean development ο στόχος οφείλει να είναι η βαθιά κατανόηση του προβλήματος του πελάτη/χρήστη και η πλήρης επίλυσή του. Όσον αφορά τη μέτρηση της ικανοποίησης του πελάτη, ο Fred Reichheld στο βιβλίο του The Ultimate Question, προτείνει ένα απλό



μέγεθος : το **Net Promoter Score**. Η ερώτηση που τίθεται στον πελάτη (μετά από την ολοκλήρωση ενός project) είναι : «Πόσο πιθανό είναι να προτείνετε σε κάποιον άλλον το προϊόν ή την υπηρεσία;». Οι πελάτες απαντούν με ένα νούμερο της κλίμακας από 0 (δεν προτείνεται) έως το 10 (προτείνεται απολύτως). Οι απαντήσεις συγκεντρώνονται ως εξής : 9-10 “Promoter”, 0-6 “Detractor”, 7-8 “Passive customer”.

Ο υπολογισμός του Net Promoter Score, γίνεται με την αφαίρεση του ποσοστού των detractors από το ποσοστό των promoters. Η τιμή που προκύπτει βρίσκεται ανάμεσα στο -100% και το 100%. Οι μέτριοι οργανισμοί λαμβάνουν ένα score ~ 10%. Οι πραγματικά καλοί οργανισμοί λαμβάνουν score 50% και άνω. Οι εταιρείες που λαμβάνουν αρνητικές τιμές στο NPS, πρέπει να επανεξετάσουν σοβαρά τις λειτουργίες και τη δράση τους.

Παράδειγμα υπολογισμού NPS.

Ερωτηθέντες	:	100
Promoters (9-10)	:	30 %
Passive (7-8)	:	50 %
Detractors (1-6)	:	20 %

Άρα το  $NPS = 30 - 20 = 10$ . ( Προφανώς πρέπει να γίνουν ενέργειες βελτίωσης από την εταιρεία).

### 8.5 Βασικά σημεία Lean μετασχηματισμού στο SW Development.

Στη συνέχεια, ολοκληρώνοντας τη διαδρομή μας, κάνουμε μια ανακεφαλαίωση πάνω στις βασικές Lean αρχές για το SW Development:

#### Optimize the Whole.

- Πρέπει να γίνεται αναγνώριση και βελτίωση του συνολικού value stream και όχι επιμέρους τμημάτων. Είναι απαραίτητη η ανάθεση της ευθύνης σε έναν Value stream owner, ο οποίος θα εντοπίσει τα κενά ή τα loops στις διαδικασίες και θα «τρέξει» τη διαδικασία βελτίωσης του Value stream.
- Αντίστοιχα πρέπει να καθοριστούν οι κατάλληλες μετρήσεις. Οι μετρήσεις πρέπει να αφορούν το «όλον» και όχι επιμέρους τμήματα/βήματα. Βασικά μεγέθη που θα μπορούσαν να μετρηθούν είναι ο **χρόνος κύκλου**, η **οικονομική απόδοση** και η **ικανοποίηση του πελάτη**, όπως είδαμε παραπάνω.
- Εξάλειψη/μείωση των λειτουργικών σιλό ανάμεσα στα τμήματα. Οι μεγαλύτερες καθυστερήσεις λαμβάνουν χώρα, όταν στο Value stream συμμετέχουν περισσότερα του ενός τμήματα.

#### Σεβασμός στους Ανθρώπους.

- Οι team leaders και οι supervisors πρέπει να λαμβάνουν την κατάλληλη εκπαίδευση και καθοδήγηση ώστε να εφαρμόζουν τις lean αρχές.
- Οι εργάτες πρώτης γραμμής, οι οποίοι και παράγουν αξία (στο SW development οι προγραμματιστές, αλλά σε συνεργασία με αναλυτές και consultants), πρέπει να έχουν την ελευθερία να σχεδιάζουν οι ίδιοι τις διαδικασίες που θεωρούν καλύτερες για τη δουλειά τους, πάντα βέβαια με την κατάλληλη επίβλεψη και σε συνεννόηση με τους ιεραρχικά ανώτερους. Οι προϊστάμενοι θα πρέπει να αναμένουν και να επιζητούν τις προτάσεις βελτίωσης. Αυτές οι προτάσεις, μετά από σχετική συζήτηση/αξιολόγηση, θα πρέπει άμεσα να υλοποιούνται.
- Δημιουργία εργασιακού κλίματος που να κάνει τους ανθρώπους να νιώθουν υπερηφάνεια για το έργο τους, για τη δουλειά τους. Τακτοποιημένοι, ευρύχωροι και ήσυχοι χώροι, καλή οργάνωση χρόνου ώστε να μην συνθλίβονται από αδύνατες προθεσμίες (deadlines), επαρκής χρόνος για ελέγχους και

αυτοματοποίηση διαδικασιών ρουτίνας είναι μερικές πρακτικές που βοηθάνε προς αυτήν την κατεύθυνση.

### **Ταχύτητα στην υλοποίηση/ Καθυστέρηση δέσμευσης.**

- Η δουλειά να γίνεται σε μικρές παρτίδες. Μείωση του χρόνου που ανεβαίνουν οι αλλαγές στην παραγωγή ανάλογα βέβαια με τις ανάγκες. Ξεκαθάρισμα όλων των λιστών/ουρών που θεωρούνται σπατάλη στο lean. (π.χ. Οι ουρές με τα “User Action” ).
- Παραλαβή νέας εργασίας μόνο εφόσον έχουν ολοκληρωθεί οι προηγούμενες ή με έναν σταθερό (όσο γίνεται) ρυθμό. Δεν είναι ορθό να «πνίγεται» η ομάδα με αιτήματα που δεν μπορεί να υλοποιήσει. Είναι δύσκολο αλλά ο Team Leader πρέπει να λέει και «Όχι».
- Ο βασικός στόχος, που δίνει και αξία στον πελάτη, είναι η μείωση του χρόνου που μεσολαβεί από την παραλαβή του αιτήματος έως την πλήρη επίλυσή του.
- Δεν είναι απαραίτητο ένα πλήρες Blueprint με τελικές προδιαγραφές για να ξεκινήσει το development. Η ανάπτυξη μπορεί να γίνεται παράλληλα και έτσι να χτίζονται και οι προδιαγραφές(πράγμα που εκ πείρας, γίνεται ούτως ή άλλως). Βέβαια δεν εννοούμε να μην υπάρχει καμιά προδιαγραφή στο ξεκίνημα ενός project, αλλά είναι πολύ φυσιολογικό στην πορεία της υλοποίησης να προκύπτουν νέες απαιτήσεις ή αλλαγές των υπαρχόντων, οι οποίες θα πρέπει να συμπεριλαμβάνονται στο έργο.
- Όταν υπάρχουν επιλογές και δεν είναι ξεκάθαρη εξ' αρχής η απόφαση που πρέπει να ληφθεί (πράγμα που συμβαίνει στις περισσότερες περιπτώσεις), πρέπει η απόφαση αυτή να λαμβάνεται όσο πιο αργά γίνεται ώστε να υπάρχουν όλες οι απαραίτητες πληροφορίες για την ορθότερη λήψη της. Είναι αυτονόητο ότι πρέπει να επενδύεται χρόνος και σκέψη σε όλες τις πιθανές λύσεις και τις συνέπειές τους, ώστε να προλαμβάνονται λάθη και παραλείψεις.(proactive φιλοσοφία)

### **Δημιουργία γνώσης/Ποιότητας.**

- Ο οργανισμός πρέπει να καλλιεργεί πνεύμα μάθησης και συνεχούς βελτίωσης. Με τον όρο μάθηση εννοούμε βεβαίως την εκπαίδευση για τη βελτίωση του επιπέδου των εργαζομένων, αλλά και τη γνώση που παράγεται από την ίδια τη δουλειά και πρέπει να μοιράζεται. Σε IT περιβάλλοντα όπως τα εταιρικά που καλούνται να υποστηρίξουν χρήστες, είναι πολύ ουσιαστική η καταγραφή τεκμηρίωσης σε απλή μορφή και με λέξεις κλειδιά, που θα μπορούσε κάποιος να αναζητήσει στα εταιρικά Intranets.
- Δεν είναι φρόνιμο να γίνονται ταυτόχρονα πολλές αλλαγές. Είναι ορθότερο να γίνονται οι αλλαγές σταδιακά ώστε να προλαβαίνει να αφομοιωθεί από τους εμπλεκόμενους η γνώση που προκύπτει από αυτές.
- Ο κώδικας πρέπει να είναι πλήρης, ορθός, ξεκάθαρος και εύκολα διαχειρίσιμος. Δεν πρέπει να μένει WIP (Work In Process). Όταν εντοπίζεται πρόβλημα πρέπει να διορθώνεται ριζικά και όχι πυροσβεστικά.
- Να γίνεται χρήση αυτοματισμών, ειδικά σε διαδικασίες ρουτίνας. Πάντα όμως με τη λογική της βελτίωσης και απλοποίησης των διαδικασιών.

### **Εξάλειψη της σπατάλης.**

- Πρέπει να υπάρχει συνεργασία IT και Business υπευθύνων, ώστε να είναι πλήρως κατανοητή η ανάγκη και να εντοπίζεται η καλύτερη λύση.
- Με κέντρο τον πελάτη/χρήστη πρέπει να εξαλείφονται οι σπατάλες από το value stream. Η προσπάθεια βελτίωσης να είναι συνεχής και συνεπής.
- Ο καλύτερος τρόπος εξάλειψης της σπατάλης είναι η μείωση των γραμμών κώδικα. Να μένουν μόνο οι απαραίτητες λειτουργίες που δημιουργούν αξία και να μειώνεται η πολυπλοκότητα.
- Σπατάλη στις επιχειρησιακές διαδικασίες είναι και οι επικοινωνιακές, όπως τα πολλά e-mail (χωρίς ορθό περιεχόμενο, με πολλούς ουσιαστικά άσχετους παραλήπτες), οι μη αποτελεσματικές συναντήσεις κλπ. Ειδικά στο θέμα της επικοινωνίας προτείνεται η ιδέα του καθημερινού 15-λεπτου standup meeting της ομάδας, το οποίο αφορά τα τρέχοντα ζητήματα.

## 8.6 Επίλογος.

Όπως έχουμε ήδη πει, το Lean είναι φιλοσοφία και κουλτούρα. Δεν είναι «μασημένη τροφή» που ο καθένας μπορεί να μιμηθεί ή να αντιγράψει. Κάθε οργανισμός πρέπει μόνος του να ανακαλύψει πώς και πού μπορεί να εφαρμόσει όσα υποστηρίζει και πρεσβεύει το Lean. Ειδικά στο SW development και στο IT, ένας πραγματικά lean μετασχηματισμός θα περιλαμβάνει απαραίτητως συνεργασία IT και Business και συνεχή προσπάθεια επίλυσης των προβλημάτων που ανακύπτουν. Ο μετασχηματισμός ξεκινάει με ανθρώπους, διαδικασίες και τεχνολογία, με αυτήν ακριβώς τη σειρά. (S.Bell-M.Orzen, 2011)

## ΚΕΦΑΛΑΙΟ 9. CASE STUDY

### 9.1 Εισαγωγή.

Στο παρόν κεφάλαιο θα παρουσιάσω τη μελέτη καθώς και τις προτάσεις μου για την εφαρμογή του Lean στο Τμήμα IT Επιχειρησιακών Εφαρμογών SAP μιας εμπορικής εταιρείας. Από τα εργαλεία που χρησιμοποιεί το Lean θα χρησιμοποιήσω τα: Kaizen, Value Stream Map, 5-Why και την εξάλειψη της σπατάλης.

Εστιάζω στην παραπάνω προσέγγιση διότι θεωρώ ότι είναι η πεμππουσία της εφαρμογής του Lean σε ένα τμήμα παροχής υπηρεσιών.

Επίσης θα προσπαθήσω να εφαρμόσω τις αρχές που διέπουν τη Lean ανάπτυξη λογισμικού που αναπτύχθηκε στο κεφάλαιο 4, συνδέοντάς τες με τα προβλήματα που αντιμετωπίζει το τμήμα και τις σπατάλες οι οποίες προκαλούν αυτά τα προβλήματα και αναπτύχθηκαν στο κεφάλαιο 6.

Η όλη διαδικασία εφαρμογής του Lean, θα πρέπει να ενταχτεί σε ένα ευρύτερο Project που θα στοχεύει τόσο στην εξάλειψη της σπατάλης όσο και στην εδραίωση της Lean κουλτούρας μέσω Kaizen.

Οι στόχοι της μελέτης είναι ο εντοπισμός των πραγματικών προβλημάτων και η πρόταση Lean λύσεων, που βασίζονται στην θεωρία του Lean όπως έχει αναλυθεί στα προηγούμενα κεφάλαια, με σκοπό την ουσιαστική βελτίωση των διαδικασιών του Τμήματος, καθώς και την αύξηση της αξίας που παράγει το συγκεκριμένο Τμήμα για τον πελάτη/χρήστη. Οι βελτιώσεις θα έχουν αντίκτυπο τόσο στην ταχύτητα και την ποιότητα εξυπηρέτησης των πελατών/χρηστών όσο και στο ηθικό των εργαζομένων στο τμήμα.

Η μελέτη του συγκεκριμένου Τμήματος και των προβλημάτων που αντιμετωπίζει έγινε μέσω της μακροχρόνιας προσωπικής παρατήρησης (λόγω εργασίας μου σε αυτό), καθώς και μέσω ανεπίσημων συνεντεύξεων με συναδέλφους τόσο από το IT όσο και από το Business.

Θα αναφέρω τη λειτουργία και την οργάνωση του Τμήματος, τη μορφολογία των χρηστών και τις υπάρχουσες διαδικασίες.

Στη συνέχεια θα παρουσιάσω τις προτάσεις μου τόσο όσον αφορά το Kaizen όσο και στην εξάλειψη της σπατάλης, παρουσιάζοντας υπαρκτά προβλήματα τα οποία έχω ήδη

εντοπίσει καθώς και τις προτεινόμενες λύσεις κι ένα ρεαλιστικό χρονοδιάγραμμα εφαρμογής τους.

Τέλος, θα προτείνω KPIs για την παρακολούθηση της εφαρμογής του Lean και θα ολοκληρώσω με τα τελικά συμπεράσματα.

## ΥΠΑΡΧΟΥΣΑ ΚΑΤΑΣΤΑΣΗ

### 9.2 Οργάνωση / Λειτουργία.

Το Τμήμα IT Επιχειρησιακών Εφαρμογών SAP της εταιρείας υποστηρίζει τους χρήστες της εταιρείας οι οποίοι δουλεύουν το γνωστό εμπορικό ERP πακέτο SAP.

Το Τμήμα είναι χωρισμένο σε υπο-τμήματα και κάθε υπο-τμήμα κάνει υποστήριξη συγκεκριμένων modules του SAP. Το SAP ως εφαρμογή είναι χωρισμένο σε modules, παρόλο που φυσικά όλα συνδέονται μεταξύ τους και λειτουργούν ολοκληρωμένα. Για την πληρέστερη κατανόηση θα κάνω μια σύντομη αναφορά στα modules του SAP που λειτουργούν στη συγκεκριμένη εταιρεία.

SD - Sales and Distribution. Ασχολείται με πωλήσεις- παραδόσεις-τιμολογήσεις

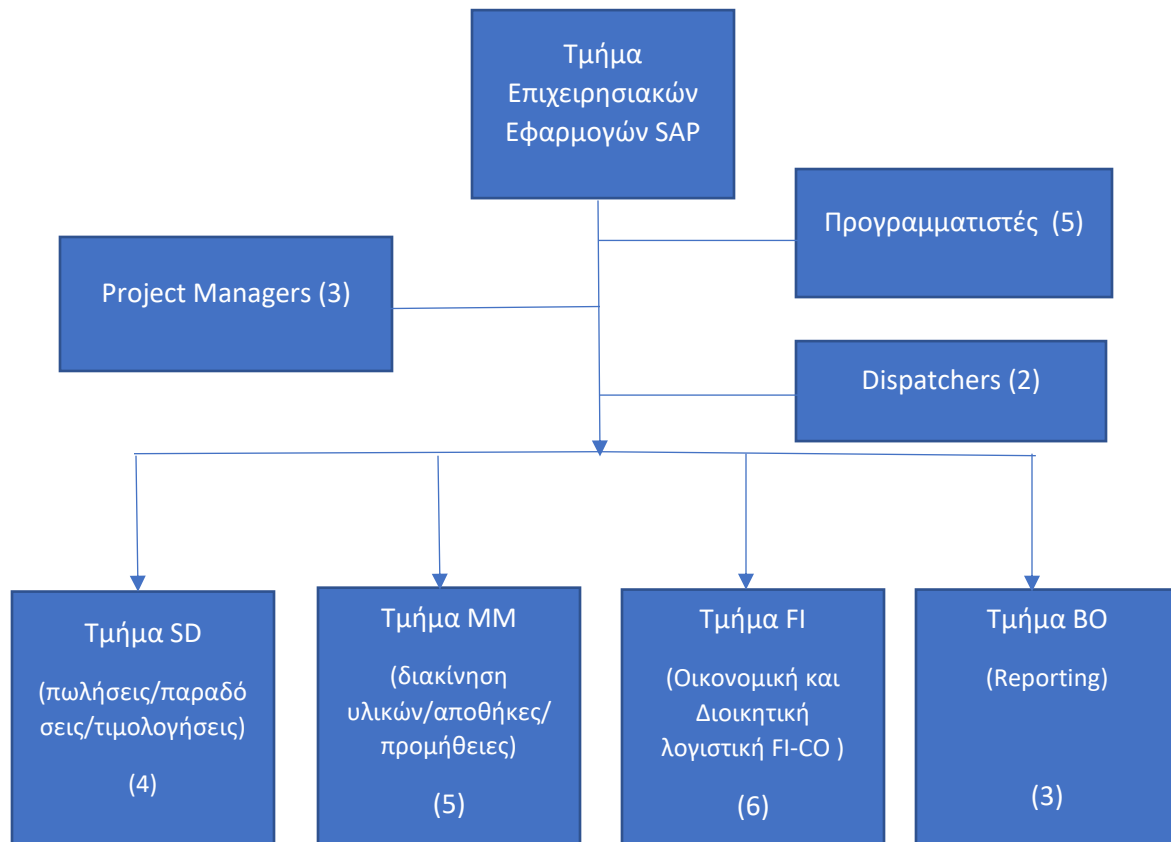
MM – Materials Management. Ασχολείται με διακίνηση υλικών, αποθήκες, προμήθειες.

FI – Financials. Η Λογιστική αποτύπωση των κινήσεων και η καταγραφή στα υποχρεωτικά από το νόμο βιβλία και καταστάσεις. ( Έκδοση οικονομικών αποτελεσμάτων, ισολογισμών κλπ.)

CO – Controlling. Η επιχειρησιακή οπτική των οικονομικών αποτελεσμάτων. Ουσιαστικά καταλήγει στο λεγόμενο «Reporting προς τη διοίκηση».

BO – Business Objects. Χρησιμοποιούνται για reporting με άντληση πληροφοριών από το ERP. Ιδιαίτερως χρήσιμα για μεγάλο όγκο δεδομένων.

Η οργάνωση του τμήματος φαίνεται παρακάτω:



Σχήμα 12. Οργανόγραμμα Τμήματος

Οι συνάδελφοι που αναφέρονται στα Τμήματα είναι οι consultants(σύμβουλοι) των modules. Η ομάδα των προγραμματιστών είναι κοινή για όλα τα modules. Οι dispatchers είναι οι συνάδελφοι που «μοιράζουν» τα αιτήματα στα υπο-τμήματα. Παράλληλα έχουν και άλλες αρμοδιότητες της Διεύθυνσης, ανεξάρτητες από το συγκεκριμένο SAP Τμήμα.

Η υποστήριξη που παρέχεται αφορά τόσο σε προβλήματα που αντιμετωπίζουν οι χρήστες, όσο και νέες απαιτήσεις και υλοποιήσεις.

Οι χρήστες αποστέλλουν το αίτημά τους (incident) μέσω του SAP ή του εταιρικού portal και ο αρμόδιος συνάδελφος (dispatcher) κρίνει σε ποια περιοχή ανήκει το αίτημα ώστε να κάνει την ανάθεση στην κατάλληλη ομάδα.

Σε περιπτώσεις που η υλοποίηση αφορά project, το αίτημα το διαχειρίζεται ο αρμόδιος Project Manager. Τα projects είναι συνήθως «κλειστά» δηλ. καθορίζεται ο χρόνος ολοκλήρωσής τους εκ των προτέρων, όπως επίσης και το scope τους. Η ομάδα



υλοποίησης μπορεί να είναι από τους εσωτερικούς πόρους ή να γίνει ανάθεση σε εξωτερικούς συνεργάτες.

Η ανάλυση του προβλήματος γίνεται από τους consultants (συμβούλους), κι όταν το πρόβλημα/ένισχυση αφορά όχι μόνο την παραμετροποίηση του συστήματος αλλά και πρόγραμμα, κατευθύνουν κατάλληλα τους προγραμματιστές. Σε ένα τμήμα που υποστηρίζει μια εφαρμογή SAP, η ιδεατή κατάσταση θα ήταν να μην χρειαζόταν ανάπτυξη κώδικα και να χρειαζόνταν μόνο οι σύμβουλοι που θα υλοποιούσαν παραμετροποιήσεις. Ακόμα πιο ιδανικά δε θα έπρεπε να υπάρχουν ούτε σύμβουλοι, αλλά key users που θα είχαν εκπαιδευτεί κατάλληλα ώστε να παίζουν το ρόλο του συμβούλου. Από την εμπειρία μου θεωρώ ότι αυτό το σενάριο είναι αρκετά δύσκολο. Το SAP ως εφαρμογή είναι αρκετά περίπλοκο και απαιτεί πειθαρχία από τους χρήστες. Επίσης μπορεί να μη φαίνεται ιδιαίτερα user friendly. Εκτός αυτών όμως πολλές φορές οι απαιτήσεις των χρηστών είναι τέτοιες που τελικά χρειάζεται η ανάπτυξη κώδικα (Z-προγράμματα). Αντίστοιχα χρειάζεται και η υποστήριξη συμβούλων, τόσο για νέες παραμετροποιήσεις όσο και για καθοδήγηση των προγραμματιστών.

### 9.3 Χρήστες.

Συνολικά οι χρήστες που δουλεύουν το SAP και είναι οι «εν δυνάμει πελάτες» του Τμήματος, είναι ~ 1500. Από την ανάλυση που έχω κάνει έως τώρα έχουν δημιουργηθεί αιτήματα από 600 χρήστες. Είναι άνθρωποι με διάφορα επίπεδα γνώσης του SAP και διάφορα επίπεδα μόρφωσης. Η εταιρεία έχει θυγατρικές στο εξωτερικό, άρα τα τμήματα υποστηρίζουν συναδέλφους και σε χώρες εκτός Ελλάδας, με επικοινωνία στα Αγγλικά.

Υπάρχει αρκετή κινητικότητα από τη μεριά των χρηστών, υπό την έννοια ότι υπάρχουν αποχωρήσεις, νέες προσλήψεις, rotation κλπ. Επίσης είναι συχνό το φαινόμενο να αποχωρούν συνάδελφοι χωρίς να έχει ολοκληρωθεί η εκπαίδευση των επόμενων, που σημαίνει αυξημένο φόρτο στο Τμήμα Επιχειρησιακών Εφαρμογών SAP, έως να φτάσει ο νέος χρήστης σε ικανοποιητικό επίπεδο γνώσης του συστήματος.

## 9.4 Υπάρχουσες διαδικασίες.

### a) ανάθεση incident σε εξωτερικό συνεργάτη

- αναγνώριση ανάγκης από το ενδιαφερόμενο τμήμα
- Επικοινωνία του συγκεκριμένου τμήματος με σύμβουλο ή εταιρεία.
- Ανάθεση της συγκεκριμένης εργασίας στον εξωτερικό σύμβουλο.
- Εφόσον ολοκληρωθεί η εργασία, αποστέλλεται τιμολόγιο για πληρωμή της υπηρεσίας.

### b) ανάθεση incident στην εσωτερική ομάδα υποστήριξης

- Ο χρήστης αποστέλλει αίτημα μέσω του SAP στο αυτόματο σύστημα (Solution Manager), με περιγραφή του προβλήματός του καθώς και το impact στην επιχειρησιακή λειτουργία. Βάσει του impact και της SLA (Service Level Agreement), καθορίζεται ο χρόνος ολοκλήρωσης, π.χ. τα High Urgency θα πρέπει να ολοκληρωθούν εντός τριών ωρών.(συνήθως τέτοιου είδους επείγοντα έρχονται τηλεφωνικά και στη συνέχεια δημιουργείται το incident)
- Ο Dispatcher εκτιμά την περιοχή στην οποία αντιστοιχεί το Incident (FI-CO-SD-MM-BO) και το αναθέτει στον αρμόδιο module leader.
- Για περιπτώσεις που δεν είναι απλή αλλαγή/διόρθωση ο module leader επικοινωνεί με το χρήστη ώστε να αντιληφθεί την έκταση του προβλήματος
- Το επόμενο βήμα είναι η ανάθεση του Incident και η υλοποίηση της συγκεκριμένης αλλαγής από την ομάδα υποστήριξης.
- Εφόσον η αλλαγή υλοποιηθεί, δίδεται για έλεγχο από το χρήστη (UAT).
- Ο χρήστης ελέγχει και αποστέλλει OK για παραγωγή ή παρατηρήσεις για διόρθωση κλπ.
- Μετά το τελικό OK του χρήστη, η αλλαγή μεταφέρεται στο παραγωγικό σύστημα.

### c) ανάθεση Project

- Αναγνώριση ανάγκης υλοποίησης κάποιου project (εκτεταμένης αλλαγής/νέας υλοποίησης).
- Η ομάδα υποστήριξης κάνει μια προσφορά (estimation) κόστους-χρόνου, βασισμένη στις αρχικές προδιαγραφές.
- Εφόσον ο χρήστης (Business Owner) αποδεχτεί την προσφορά, γίνεται λεπτομερής καταγραφή των προδιαγραφών (Blueprint), και ορίζεται ο χρόνος παράδοσης για UAT και τελικής παράδοσης του έργου.

#### **d) Διαδικασία εγκρίσεων.**

Για τις αλλαγές που μεταφέρονται στην παραγωγή, υπάρχει μια διαδικασία εγκρίσεων που στην παρούσα φάση έχει σημεία σπατάλης. Ζητείται η έγκριση του Change Manager πριν ξεκινήσει η αλλαγή και του module leader πριν περάσει στην παραγωγή.

Η διαδικασία αυτή έχει συμφωνηθεί με τους auditors ώστε να υπάρχει έλεγχος δύο σημείων για τις αλλαγές που περνάνε στην παραγωγή ώστε να είναι πλήρως διασφαλισμένες και ελεγμένες.

Το πρόβλημα είναι ότι στην παρούσα φάση δε γίνεται κάποιος ιδιαίτερος έλεγχος στο πρώτο σημείο, άρα με τον υφιστάμενο τρόπο λειτουργίας είναι ξεκάθαρη σπατάλη και θα μπορούσε ή να αναβαθμιστεί και να γίνει πιο ουσιαστικός ή να παραληφθεί.

## **ΠΡΟΤΑΣΕΙΣ ΕΦΑΡΜΟΓΗΣ LEAN**

### **9.5 Kaizen Initialization Project**

Το πρώτο βήμα για την εφαρμογή του Lean στο Τμήμα, θα πρέπει να είναι ένα project για την εγκατάσταση ενός συστήματος Kaizen με εξειδικευμένους Lean συμβούλους. Με τον τρόπο αυτό θα γίνει ουσιαστική γνωριμία των συναδέλφων με το Lean και θα τεθούν οι βάσεις για τη συνέχεια. Θα πρέπει να εμπλακούν όλοι οι συνάδελφοι, ώστε :

- να γίνει ξεκάθαρος ο στόχος σε όλους, που είναι η βελτίωση των διαδικασιών, η μείωση της σπατάλης και η αύξηση της αξίας για τον πελάτη μας (χρήστη).
- να αισθάνονται οι συνάδελφοι πως κανείς δεν περισσεύει, ότι είναι απαραίτητοι στο έργο και ότι η Διεύθυνση πραγματικά σέβεται του ανθρώπους της.
- να έρθουν οι συνάδελφοι σε επαφή με τις μεθόδους και τα εργαλεία του Lean

Με το αρχικό αυτό project, θα βελτιωθούν τα βασικά value streams και οι υπάρχουσες διαδικασίες και με την κατάλληλη εστίαση σε early wins (γρήγορες νίκες), θα υπάρχουν άμεσα αποτελέσματα που θα βοηθήσουν στην ενσωμάτωση της Lean κουλτούρας στο τμήμα, στην ενίσχυση της ομαδικότητας και στη δημιουργία ευχάριστου εργασιακού κλίματος, διότι πλέον όλο το Τμήμα θα έχει ένα κοινό στόχο και θα παλεύει για αυτόν.

Επίσης με το αρχικό αυτό project θα δοθεί έμφαση και χρόνος στην κατανόηση των βαθύτερων αιτιών των προβλημάτων από όλους τους συναδέλφους, διότι ως γνωστόν η

καθημερινότητα οδηγεί στο «δεν προλαβαίνω να δώσω χρόνο στο σχεδιασμό και στην εύρεση και διόρθωση των πραγματικών αιτιών».

### **9.6 Καθημερινό Kaizen.**

Αυτό το στάδιο είναι η πεμπτουσία βεβαίως του Lean, αλλά και το πλέον δύσκολο. Αφορά όπως γνωρίζουμε τη συνεχή βελτίωση των διαδικασιών και απευθύνεται τόσο στις ομάδες όσο και ατομικά σε κάθε συνάδελφο.

Όπως έχουμε ήδη πει, πολλές προσπάθειες lean μετασχηματισμού αποτυγχάνουν, διότι οι άνθρωποι επιστρέφοντας από το αρχικό Kaizen Initialization Project, επανέρχονται στις παλιές κακές συνήθειες. Το Kaizen που βασίζεται μόνο στο αρχικό project είναι καταδικασμένο να αποτύχει.

Το καθημερινό kaizen που προτείνω είναι η βασική lean μέθοδος του **stop-the-line**. Όσο δύσκολο και ουτοπικό κι αν ακούγεται στο συγκεκριμένο τμήμα, είναι η μόνη μέθοδος που μπορεί να επιφέρει αποτελέσματα. Για να είμαστε όμως ρεαλιστές δεν μπορεί μια εταιρεία να σταματήσει τη δουλειά της (να μην πουλάει, να μη βγάλει αποτελέσματα κλπ.), επειδή «έσκασε» ένα πρόγραμμα και δεν μπορούν π.χ. να τυπωθούν τιμολόγια ή να γίνει το μηνιαίο κλείσιμο, έως ότου βρεθεί η πραγματική αιτία που είναι π.χ. η εξάντληση του προγραμματιστή ή η κακή διαδικασία του UAT. Σε τέτοιες περιπτώσεις προφανώς πρέπει να δοθεί μια πυροσβεστική λύση. Το stop-the-line θα έχει την έννοια του να γίνει γνωστό το πρόβλημα ώστε να μπει στη διαδικασία εύρεσης της πραγματικής αιτίας.

Για να προσαρμόσουμε το Lean στα μέτρα μας και στη δική μας πραγματικότητα, προτείνω κατά το καθημερινό kaizen, την ανεύρεση και αντιμετώπιση των πραγματικών αιτιών σε δεύτερο χρόνο που θα καθορίζεται άμεσα από τους εμπλεκόμενους, ώστε να αποφευχθεί ο κίνδυνος να αφηθεί η οποιαδήποτε βελτίωση στη μοίρα της και να επανέλθουμε στα παλιά.

### **9.7 Ορισμός υπευθύνου για το καθημερινό Kaizen.**

Κρίνω απαραίτητη την ανάθεση του συντονισμού και του ελέγχου του καθημερινού kaizen σε έναν υπεύθυνο που θα ενστερνίζεται πραγματικά τη Lean κουλτούρα. Ο ρόλος του θα είναι ο έλεγχος της βελτίωσης των διαδικασιών, να μην «ξεχνιούνται» εντοπισμένα προβλήματα, να ενισχύει τις ομάδες και τους συναδέλφους, έτσι ώστε να διατηρείται ζωντανή η κουλτούρα του Lean και να εξελίσσεται το Τμήμα σε learning organization.

Ο Kaizen υπεύθυνος θα πρέπει να αναφέρεται κατευθείαν στο Διευθυντή Πληροφορικής ώστε να είναι ξεκάθαρο σε όλους ότι είναι σημαντικός ο ρόλος του κι ότι το Lean έχει νόημα και σημασία για το Τμήμα. Εδώ βέβαια πρέπει να τονιστεί η τεράστια σημασία της Ηγεσίας. Όπως έχουμε δει και στη θεωρία, εάν η Διοίκηση δεν ασπάζεται πραγματικά την έννοια του Lean, ο μετασχηματισμός θα αποτύχει.

### **9.8 Εντοπισμός Βασικών Προβλημάτων - Εντοπισμός Σπαταλών - Προτεινόμενες Λύσεις**

Μέσα από τη μελέτη των αιτημάτων και της λειτουργίας του Τμήματος, έχω καταλήξει σε κάποια πολύ σημαντικά προβλήματα, τα οποία επιδεινώνουν τη λειτουργία του. Με βάση όσα έχουμε πει θα μπορούσαμε να δημιουργήσουμε και να μελετήσουμε τα value streams του τμήματος. Όμως θεωρώ πως στη φάση αυτή δε χρειάζεται η απεικόνιση των value streams για να εντοπιστούν οι σπατάλες, διότι αυτό είναι από μόνο του ένα ουσιαστικό project.

Είναι όμως εφικτό να ξεκινήσουμε με τις βασικότερες σπατάλες οι οποίες είναι ήδη εντοπισμένες. Τα προβλήματα θα αναγνωριστούν/κατηγοριοποιηθούν βάσει των επτά βασικών σπαταλών στο software development, όπως καταγράφονται στο κεφάλαιο 6. Όπως έχει ήδη αναφερθεί, η κατηγοριοποίηση δεν είναι αυτοσκοπός, αλλά ένα μέσο διερεύνησης και αποκάλυψης της σπατάλης, ώστε τελικά να εξαιρεθεί. Παρατίθενται βάσει της ανάλυσης **5-Whys** (βασική τεχνική ανάλυσης ώστε να φτάσουμε στην πραγματική πηγή του προβλήματος ρωτώντας «Γιατί» 5 φορές) τόσο οι σπατάλες όσο και ο τρόπος αντιμετώπισης, καθώς και οι Lean αρχές που ικανοποιούν τις συγκεκριμένες λύσεις και έχουν αναλυθεί στο κεφάλαιο 4.

Τα παρακάτω προβλήματα, οι προτεινόμενες λύσεις και το χρονοδιάγραμμα, θα ενταχθούν στο Kaizen Initialization Project ώστε να εξευρεθούν οι πρώτες λύσεις και βελτιώσεις διαδικασιών οι οποίες θα αποτελέσουν τα early wins. Επειδή όμως αφορούν πολύ κρίσιμες λειτουργίες του Τμήματος, θα αποτελέσουν αντικείμενο και του Καθημερινού Kaizen.

### 9.8.1 Απροσδόκητος τερματισμός λειτουργίας.

Υπάρχει δηλ. αδυναμία του συστήματος να ανταποκριθεί, με αποτελέσματα σημαντικά τόσο για τους χρήστες όσο και για την εταιρεία.

#### Γιατί (1);

- Αστοχία στον κώδικα.

#### Γιατί (2);

- Υπάρχει πολυπλοκότητα στα προγράμματα. Πολλές φορές είναι δύσκολο να γίνουν αλλαγές, λόγω δυσχερούς και κακογραμμένου προϋπάρχοντος κώδικα.
- Ελλιπής έλεγχος. Οι αλλαγές που γίνονται, ιδιαίτερα σε κώδικα, ελέγχονται πλημμελώς, τόσο από τους developers όσο και από τους χρήστες.
- Κακός κώδικας λόγω επαγγελματικής άγνοιας του προγραμματιστή.

#### Γιατί (3); (γιατί δεν «πιάστηκαν» τα λάθη έγκαιρα κατά τη φάση ανάπτυξης;)

- δεν υπήρχαν αρκετά σενάρια
- οι έλεγχοι έγιναν μόνο στα προφανή σενάρια

**Διάγνωση Σπατάλης:** «Ελαττωματικός κώδικας (Defects)»

**Lean Αρχή Προτεινόμενης Λύσης:** «Καθυστέρηση Δέσμευσης/Δημιουργία Ποιότητας»

#### Περιγραφή Λύσης:

- Για τη βελτίωση του κώδικα αλλά και γενικότερα των προσφερόμενων λύσεων, πρέπει να ακολουθείται ο κύκλος PDCA(Plan-Do-Check-Act) με αυξημένο χρόνο στο Planning. Επίσης πρέπει να ακολουθείται η βασική αρχή της «Αναβολής δέσμευσης (Defer Commitment)». Είναι σημαντικό να υπάρχει χρόνος μελέτης του προβλήματος, ώστε να επιλεγεί/ επιτευχθεί η καλύτερη δυνατή λύση.
- Τα specifications δημιουργίας/αλλαγής κώδικα θα πρέπει να εξετάζονται με προσοχή από τους συμβούλους και τους προγραμματιστές, ώστε να αποφεύγεται επαναλαμβανόμενος κώδικας. (π.χ. χρήση Functions σε κοινά προγράμματα αντί της ίδιας αλλαγής στα προγράμματα ).
- Καλή πρακτική που απαντάται και στη διεθνή βιβλιογραφία είναι το pair programming. Ο κώδικας ελέγχεται από συνάδελφο, πράγμα που βοηθάει στη συνεχή βελτίωση του κώδικα αλλά και των προγραμματιστών.

- Τακτική (αλλά και έκτακτη) εκπαίδευση των προγραμματιστών και των συμβούλων σε νέες εξελίξεις που αφορούν το χώρο τους.
- Προτείνεται η χρήση κοινής ονοματολογίας στα νέα προγράμματα/transactions καθώς και η χρήση comments στην αρχή του κώδικα (με αναφορά στην ημερομηνία αλλαγής, στο incident και στον προγραμματιστή)
- Για κάθε αλλαγή / νέα υλοποίηση, είναι απαραίτητη η δημιουργία σεναρίων προς έλεγχο και φυσικά ο έλεγχος αυτών των σεναρίων τόσο από τους αρμόδιους συμβούλους/προγραμματιστές αλλά και από τους χρήστες.
- Στη διαδικασία αλλαγών, δε θα πρέπει να περνάνε αλλαγές χωρίς OK των χρηστών σε όλα τα σενάρια.
- Εφόσον η προτεινόμενη αλλαγή περάσει στην παραγωγή, ο αρμόδιος χρήστης πρέπει να επιβεβαιώνει ότι η αλλαγή λειτουργεί όπως σχεδιάστηκε.

#### **Σημαντικότητα προτεινόμενης λύσης: Υψηλή.**

- Το κόστος αστοχίας κώδικα είναι πολύ υψηλό από οικονομική αλλά και ψυχολογική άποψη. Απαιτούνται αρκετές ανθρωποώρες για την εύρεση και αποκατάσταση του προβλήματος από τη μεριά του IT, καθώς και των χρηστών οι οποίοι δεν μπορούν να εκτελέσουν τις εργασίες τους. Θεωρώ επίσης ότι οι προτεινόμενες λύσεις θα βοηθήσουν στην εμπέδωση της lean κουλτούρας, διότι δίνουν βάση στην πρόληψη και όχι στη θεραπεία.

#### **Ευκολία προτεινόμενης λύσης – Μεσαία.**

- Οι παραπάνω λύσεις θεωρώ ότι είναι άμεσα υλοποιήσιμες. Απαιτούν περισσότερο χρόνο στη φάση του Planning και είναι αναμενόμενη η αντίδραση της μορφής «Δεν προλαβαίνουμε». Χρειάζεται η υποστήριξη της διοίκησης για την εφαρμογή της.

### 9.8.2 Διαφορετική εικόνα στα συστήματα Παραγωγικό/Development/QA

(σε προγράμματα / παραμετροποίηση)

#### Γιατί (1);

- Υλοποιήθηκαν αλλαγές σε objects τα οποία ήταν ήδη σε φάση αλλαγής από άλλη ομάδα υποστήριξης, λόγω κάποιου project ή άλλου incident.

#### Γιατί (2);

- Υπάρχει άγνοια από τις ομάδες υποστήριξης για τα projects που βρίσκονται σε εξέλιξη και των πιθανών κοινών περιοχών αλλαγής. Φυσικά ισχύει και το αντίστροφο. Δηλ. πιθανά μέσω ενός Project υλοποιούνται αλλαγές σε προγράμματα / παραμετροποίηση για τα οποία παράλληλα μπορεί να γίνεται και αλλαγή μέσω Incident. «Δε γνωρίζει η δεξιά τι ποιεί η αριστερά»
- Υπάρχει άγνοια ανάμεσα στις ομάδες υποστήριξης για πιθανά κοινά incidents.
- Δεν υπάρχει αποτελεσματικός έλεγχος των συμβούλων που υλοποιούν αλλαγές, με αποτέλεσμα να διαπιστώνονται μη ολοκληρωμένες υλοποιήσεις. Δυστυχώς αυτό μπορεί να ανακαλυφθεί εντελώς τυχαία.

#### Γιατί (3);

- Μεγάλη καθυστέρηση στην μεταφορά των αλλαγών στην παραγωγή, με αποτέλεσμα κάποια στιγμή να προκύψει η παραπάνω κατάσταση, δηλ. μια ομάδα να ασχολείται με ένα αντικείμενο το οποίο ήδη βρίσκεται σε UAT από άλλη ομάδα κλπ.
- Έλλειψη ελέγχου κατάστασης(status) του project από τον αρμόδιο project manager.

#### Γιατί (4);

- οι χρήστες δεν προλαβαίνουν να κάνουν ελέγχους σε αυτά που έχουν ζητήσει.
- οι project managers δεν προλαβαίνουν να κάνουν όλες τις απαιτούμενες ενέργειες.

**Διάγνωση Σπατάλης:** «Ημιτελής δουλειά (Partially Done Work)»

**Lean Αρχή Προτεινόμενης Λύσης:** «Ταχύτητα στην υλοποίηση», «Δημιουργία Γνώσης»

**Περιγραφή Λύσης:**



- Για την αντιμετώπιση της άγνοιας των διαφόρων projects και των πιθανών κοινών περιοχών με άλλα projects ή incidents, είναι απαραίτητο το visualization. Προτείνεται η χρήση πινάκων(visual boards) στους οποίους θα καταγράφεται το status των projects, με τακτική ενημέρωση(π.χ. εβδομαδιαία) από τον αρμόδιο project manager. Βέβαια οι πίνακες προϋποθέτουν την ύπαρξη κάποιου χώρου. Στο συγκεκριμένο open space χώρο θα μπορούσαν να χρησιμοποιηθούν τα διαχωριστικά των γραφείων ή κάποιο γραφείο που θα χρησιμοποιείται για αυτή τη δουλειά ως «δωμάτιο πολέμου –Obeya». Είναι πιο εύκολη η λύση ηλεκτρονικού χώρου τον οποίο θα ενημερώνουν ανάλογα οι project managers.
- Για την καλύτερη ενημέρωση των ομάδων, προτείνεται το καθημερινό 15-λεπτο standup meeting, που ουσιαστικά αφορά στο τι έγινε την προηγούμενη ημέρα και τι πρέπει να γίνει σήμερα, καθώς και πιθανά προβλήματα που έχουν ανακύψει. Η επίλυση των προβλημάτων θα γίνεται με περαιτέρω ενέργειες.
- Επίσης προτείνεται ένα εβδομαδιαίο meeting (~1,5 h) συνολικά για όλες τις ομάδες υποστήριξης καθώς και των projects, ώστε να γνωρίζουν όλοι σε ποια φάση βρίσκονται τα projects, όπως επίσης και τις επιλύσεις που έχουν δοθεί/προγραμματίζονται να δοθούν σε σημαντικά incidents.
- Θα πρέπει να υπάρχει καλύτερη επικοινωνία με το Business όσον αφορά την αναγκαιότητα των incidents που δημιουργούν και ενημέρωσή τους για τα υπάρχοντα. Ένας κανόνας θα μπορούσε να είναι η ΜΗ παραλαβή νέου incident από χρήστη που έχει ήδη δύο incidents σε εξέλιξη.

#### **Σημαντικότητα προτεινόμενης λύσης – Υψηλή.**

- Είναι απαραίτητη η γνώση ανάμεσα στις ομάδες υποστήριξης και projects. Ουσιαστικά αφορά το σπάσιμο των σιλό που δυστυχώς υπάρχουν στις εταιρείες.

#### **Ευκολία προτεινόμενης λύσης.**

- Η δημιουργία του Obeya είναι μάλλον **Δύσκολη**, διότι θα πρέπει να πειστούν αρκετοί συνάδελφοι και ειδικά η ανώτερη διοίκηση(CIO), που θα πρέπει να το επιβάλλει ως κανόνα. Όμως είναι **Εύκολη** και προσιτή η χρήση ηλεκτρονικού χώρου.
- Όσον αφορά τα meetings και τον κανόνα των 2 incidents, θεωρώ ότι είναι **Εύκολη** και άμεσα υλοποιήσιμη.

### 9.8.3 Εξουθένωση συναδέλφων/ κακό εργασιακό κλίμα.

#### Γιατί (1);

- Πολλά αιτήματα έρχονται ως επείγοντα.
- Υπάρχουν πολλά αιτήματα που περιμένουν επίλυση. Μεγάλο backlog (έως και 200 incidents για κάποιες ομάδες)
- Οι εργαζόμενοι στα projects δουλεύουν ταυτόχρονα και στο support, άρα προσπαθούν να μοιράσουν το χρόνο τους.

#### Γιατί (2);

- Οι χρήστες αναγνωρίζουν το πρόβλημα «τελευταία στιγμή».
- Οι χρήστες αφήνουν Incidents σε UAT.
- Τακτικά ελέγχονται οι λίστες των incidents για καθορισμό προτεραιότητας, υπενθυμίσεις με mail, τηλέφωνα για αυτά που είναι User Action, διαδικασία που είναι τόσο χρονοβόρα όσο και ψυχοφθόρα.

#### Γιατί (3);

- πολλά επείγοντα αφορούν «επαναλαμβανόμενα» προβλήματα, τα οποία δε λύνονται ριζικά αλλά πυροσβεστικά.
- υπάρχει κακή επικοινωνία ανάμεσα σε Business και IT. Δεν έχουν καθοριστεί επιτυχώς οι προδιαγραφές για τη ριζική επίλυση των προβλημάτων.
- Πολλές φορές οι χρήστες δημιουργούν incident, δίνοντας και τη «λύση». Δηλ. ουσιαστικά περιγράφουν πώς σκέφτονται την υλοποίηση της λύσης που σκέφτονται για το πρόβλημά τους. Όμως δεν περιγράφουν (και δεν απαιτείται από κάποια διαδικασία) το πραγματικό πρόβλημα, την πραγματική ανάγκη.
- Οι συνάδελφοι αναγκάζονται να ασχοληθούν με τα επείγοντα, αφήνοντας την τρέχουσα εργασία τους.
- ο χώρος εργασίας είναι ο open space, που δε βοηθάει καθόλου στη συγκέντρωση. Υπάρχουν φωνές, τηλέφωνα, ανοιχτές ακροάσεις που δυσχεραίνουν την εργασία.
- η διαδικασία εγκρίσεων οδηγεί σε αντιπαραθέσεις και περιττές επικοινωνίες.

**Διάγνωση Σπατάλης:** «Ημιτελής δουλειά (Partially Done Work)», «Εναλλαγή εργασίας (Task switching )»

**Lean Αρχή Προτεινόμενης Λύσης:** «Optimize the Whole», «Σεβασμός στους Ανθρώπους».

### Περιγραφή Λύσης:

- Για να μειωθεί η σπατάλη χρόνου λόγω switching, θα πρέπει τα projects να υλοποιούνται με διαφορετική ομάδα από εκείνη της υποστήριξης.
- Για τη διαπίστωση και διερεύνηση των πραγματικών αιτιών των προβλημάτων θα πρέπει οι consultants/developers να επισκέπτονται το χώρο εργασίας των χρηστών (Gemba) ώστε να παρατηρούν πώς γίνεται πραγματικά η δουλειά και πώς μπορούν να βοηθήσουν τους χρήστες στο συγκεκριμένο πρόβλημά τους. Για τα IT προβλήματα αυτό μπορεί να γίνει virtually π.χ. μέσω Skype όπου ο σύμβουλος «βλέπει» τι ακριβώς κάνει ο χρήστης στο δικό του workstation.
- Η παρατήρηση του χώρου εργασίας όμως οφείλει να γίνεται και από τους IT leaders, ώστε να αντιλαμβάνονται τις βελτιώσεις που πρέπει να υλοποιήσουν στις υπάρχουσες διαδικασίες. Οι leaders, συμπεριλαμβανομένου και του CIO, πρέπει να είναι κοντά στην καθημερινότητα των συνεργατών τους. Πρέπει να ακούνε πραγματικά και να λαμβάνουν σοβαρά υπόψιν τους συνεργάτες τους.
- Προτείνεται η χρήση του A3 για τη διερεύνηση των πραγματικών αιτιών ενός προβλήματος. Επίσης η χρήση της ανάλυσης 5-Why.
- Ο εργασιακός χώρος θα έπρεπε να είναι ήσυχος, ώστε να μπορούν ειδικά οι προγραμματιστές να εργάζονται συγκεντρωμένοι και χωρίς απόσπαση προσοχής. Το ιδανικό θα ήταν τα προσωπικά γραφεία, επειδή όμως κάτι τέτοιο είναι μάλλον απίθανο, θα πρότεινα διαχωριστικά ανάμεσα σε κάθε γραφείο καθώς επίσης χρήση του skype call και conference call για τις επικοινωνίες(αντί της ανοιχτής ακρόασης).
- Θα πρέπει η εκπαίδευση των υπαλλήλων να αφορά τόσο την ανάπτυξη/βελτίωση των τεχνικών δεξιοτήτων τους, όσο και σε γενικότερες γνώσεις (π.χ. σεμινάρια management / time management / ITIL κλπ.). Με αυτόν τον τρόπο θα αισθανθούν καλύτερα οι συνεργάτες στο IT και θα αυξηθεί τόσο η ικανοποίησή τους όσο και η απόδοσή τους.
- Είναι σημαντικό να γίνεται **coaching** στους εργαζόμενους στο Τμήμα, και να παρακολουθούν ειδικά σεμινάρια οι IT leaders, ώστε να μπορούν να ενισχύσουν την ομάδα τους μέσω του coaching και όχι βάσει του γνωστού (κακού) μοντέλου Command and Control.
- Θα πρέπει να αλλάξει η διαδικασία των εγκρίσεων μέσα από επικοινωνία με τους auditors. ( π.χ. να γίνεται ένα μόνο approve πριν τη μεταφορά στην παραγωγή, που βέβαια πρέπει να αφορά ουσιαστικό έλεγχο και όχι απλό OK)

### **Σημαντικότητα προτεινόμενης λύσης – Υψηλή.**

- Είναι πολύ σημαντικό για τους εργαζόμενους να μην «καίγονται» από την υπερπροσπάθεια. Άρα θα έπρεπε να υπάρχει διαχωρισμός ανάμεσα σε Projects και Support όσον αφορά στους πόρους.
- Επίσης απολύτως σημαντική είναι η παρακολούθηση του τι κάνει ο χρήστης, ώστε να δοθεί η ορθή λύση στο πρόβλημά του.
- Εξίσου σημαντικός είναι ο χώρος εργασίας να επιτρέπει την απρόσκοπτη εργασία, χωρίς παρεμβολές.
- Ο ρόλος της εκπαίδευσης και του coaching είναι τεράστιας σημασίας, διότι θα αναπτερώσει το ηθικό των συναδέλφων.

### **Ευκολία προτεινόμενης λύσης.**

- Θεωρώ ότι ο διαχωρισμός των πόρων ανάμεσα σε Projects και Support είναι **Δύσκολος** και δε θα ευοδωθεί τουλάχιστον άμεσα. **Δύσκολη** είναι και η εφαρμογή του A3, διότι χρειάζεται μεγάλη αλλαγή στην κουλτούρα και εκπαίδευση. Αντίστοιχα **Δύσκολη** είναι η μετατροπή του χώρου από Open Space σε κλειστά γραφεία ή σε διαχωρισμένα τμήματα.
- Είναι όμως **Εύκολη** η παρακολούθηση της δουλειάς του χρήστη ώστε να αντιμετωπίζεται το πρόβλημα στη ρίζα του καθώς και η ανάλυση 5-Why (τουλάχιστον όσον αφορά στις ερωτήσεις)
- **Εύκολη** και **απαραίτητη** είναι η συμμετοχή της ανώτερης διοίκησης του IT στη βελτίωση των εσωτερικών διαδικασιών.
- **Εύκολη** είναι και η χρήση της τεχνολογίας για τη δημιουργία ενός εργασιακού περιβάλλοντος φιλικό προς τον εργαζόμενο.
- **Μέτριας** δυσκολίας είναι η εκπαίδευση και το coaching.

#### 9.8.4 Παρενέργειες υλοποιημένων έργων.

Πολλά έργα (projects) όταν ανεβαίνουν στην παραγωγή, παρουσιάζουν προβλήματα τα οποία επιλύονται με τη δημιουργία νέων incidents. Αυτό σημαίνει ακριβότερη κοστολόγηση των έργων εφόσον εκτός του αρχικού κόστους προστίθενται και τα κόστη των σχετικών incidents. Επίσης αυξάνεται ο χρόνος που είχε δοθεί για τα projects εις βάρος των άλλων incidents. Αποτέλεσμα είναι καθυστερήσεις άλλων υλοποιήσεων, κακό κλίμα λόγω των προβλημάτων που ανακύπτουν, αναστάτωση των χρηστών που δε μπορούν να δουλέψουν, με όποια συνέπεια μπορεί να σημαίνει αυτό για μια εταιρεία. (θυμωμένοι πελάτες/χρήστες, κακή φήμη του IT και της εταιρείας, κακή ποιότητα υπηρεσίας, χαμηλό ηθικό εργαζομένων στο IT).

##### Γιατί(1);

- Οι προδιαγραφές άλλαξαν κατά την υλοποίηση, αλλά δεν μπορούσε να ακολουθήσει η υλοποίηση, διότι ήταν εκτός scope.
- Υπήρξαν σενάρια που δεν λήφθηκαν υπ' όψιν ούτε κατά την υλοποίηση ούτε κατά τους ελέγχους.

##### Γιατί(2);

- Δεν υπήρχε ο απαραίτητος χρόνος για μελέτη του project και των σεναρίων, ούτε για ελέγχους.

##### Γιατί(3);

- Δεν ήταν ρεαλιστικές οι ημερομηνίες παράδοσης.
- Δεν υπήρχαν καταγεγραμμένα σενάρια ελέγχου.
- Τόσο ο Business Owner από τη μεριά των χρηστών, όσο και ο IT Project Manager δεν είχαν στήσει ορθά το πλαίσιο του Project.

**Διάγνωση Σπατάλης:** «Ημιτελής δουλειά (Partially Done Work)»

**Lean Αρχή Προτεινόμενης Λύσης:** «Καθυστέρηση δέσμευσης», «Δημιουργία Γνώσης», «Εστίαση στην ποιότητα».

##### Περιγραφή Λύσης:

- Όπως έχει προαναφερθεί το μοντέλο του καταρράκτη (waterfall) έχει αποδειχτεί αναποτελεσματικό. Η εμμονή στο score και στο blueprint, επίσης δεν είναι λειτουργική. Η πρότασή μου είναι η καθυστέρηση της δέσμευσης ώστε να

συγκεντρωθούν όσο γίνεται περισσότερες πληροφορίες και η τμηματοποίηση της υλοποίησης. Να «σπάει» το έργο σε φάσεις και αυτές οι φάσεις να περνάνε στην παραγωγή, ώστε οι χρήστες να προλαβαίνουν να επεξεργάζονται τις νέες αλλαγές και να βλέπουν πιθανά προβλήματα.

- Επίσης όμως πρέπει να δίνεται μεγάλη σημασία στην ποιότητα της υλοποίησης, ειδικά στην ορθότητα του κώδικα και στα σενάρια ελέγχου.
- Όσον αφορά τη δημιουργία γνώσης, έχει να κάνει με το πλήρες documentation που πρέπει να παραδίδεται με κάθε νέο Project. Θεωρώ φρόνιμο και σύμφωνα με τη Lean κουλτούρα το documentation να είναι συνοπτικό και περιεκτικό με τη μορφή σεναρίων.
- Προτείνεται επίσης η εκπαίδευση των εμπλεκόμενων Project Managers (τόσο από το Business όσο και από το IT), μέσω κατάλληλων προγραμμάτων.
- Προτείνεται η στενότερη συνεργασία Business IT ώστε να γίνεται ρεαλιστικός προγραμματισμός των αλλαγών.

#### **Σημαντικότητα προτεινόμενης λύσης – Υψηλή.**

- Είναι προφανώς μεγάλης σημαντικότητας τα επιτυχημένα projects. Έτσι αυξάνεται η αξία για τους πελάτες/χρήστες αλλά και συνολικά η αξία της εταιρείας.

#### **Ευκολία προτεινόμενης λύσης.**

- Θεωρώ την αλλαγή της κουλτούρας όσον αφορά το Scope των projects αρκετά **Δύσκολη**. Απαιτεί γενικότερη αλλαγή της κουλτούρας που αφορά όλη την εταιρεία και θα έπρεπε να εισαχθεί σε γενικότερο πλαίσιο μετασχηματισμού.
- Μπορεί όμως με **Μέτρια** δυσκολία να γίνεται η τμηματοποίηση των projects. Αυτό βέβαια απαιτεί τη στενή συνεργασία με το Business ώστε να υπάρξει συμφωνία.
- **Εύκολη** σχετικά θεωρώ την προσπάθεια για βελτίωση της ποιότητας των παραδοτέων του project.
- Επίσης **Εύκολα** μπορεί να ενταχθεί στο εταιρικό εκπαιδευτικό πρόγραμμα η σχετική εκπαίδευση.

### **9.8.5 Δημιουργία πολλών custom προγραμμάτων.**

Αυτά είναι προγράμματα που καλύπτουν ανάγκες, τις οποίες δεν μπορεί να καλύψει το standard SAP. Το αποτέλεσμα είναι αύξηση της πολυπλοκότητας του συστήματος.

#### **Γιατί(1);**

- άγνοια της ύπαρξης standard λύσης (εάν υπάρχει).
- είναι η εύκολη λύση στις απαιτήσεις των χρηστών.

#### **Γιατί(2);**

- Δε γίνεται διερεύνηση της πραγματικής ανάγκης, ώστε να ελεγχθεί εάν υπάρχει ήδη κάτι που μπορεί να καλύπτει την πρωτογενή ανάγκη του χρήστη.
- Δεν υπάρχει κάποιος τρόπος (εύκολος τουλάχιστον), να γνωρίζει κάποιος τι έχει δημιουργηθεί για συγκεκριμένες ανάγκες.

#### **Γιατί(3);**

- Δεν υπάρχει ο απαραίτητος χρόνος για διερεύνηση του πραγματικού προβλήματος, της πηγής. Επίσης δεν υπάρχει ο χρόνος για διερεύνηση πιθανής ύπαρξης standard SAP λύσης.

**Διάγνωση Σπατάλης:** «Υπερβάλλουσα λειτουργικότητα», «Επανεύρεση λύσης».

**Lean Αρχή Προτεινόμενης Λύσης:** «Optimize the Whole», «Καθυστέρηση δέσμευσης», «Δημιουργία Γνώσης».

#### **Περιγραφή Λύσης:**

- Να δίνεται αρκετός χρόνος στη διερεύνηση του πραγματικού προβλήματος.
- Να μη λαμβάνονται βιαστικά οι αποφάσεις υλοποίησης οποιασδήποτε λύσης. Να γίνεται όσο γίνεται μεγαλύτερη συγκέντρωση στοιχείων. Για τη βελτίωση της επικοινωνίας ανάμεσα σε IT και business, θα πρέπει οι χρήστες να επεξηγούν και να τεκμηριώνουν τις αλλαγές που ζητάνε με οικονομικούς όρους ή εργατοώρες κλπ. καθώς και να απαντάνε στις ερωτήσεις:
  - Ποιο είναι το πρόβλημα προς επίλυση?
  - Γιατί είναι σημαντικό ?
  - Πώς ανακαλύφθηκε το πρόβλημα ?

- Στα πλαίσια της δημιουργίας γνώσης προτείνεται η δημιουργία documentation που να αφορά και διαδικασίες, με χρήση λέξεων κλειδιών που θα βοηθήσει στην ανεύρεση πιθανής υπάρχουσας λύσης.

#### **Σημαντικότητα προτεινόμενης λύσης – Υψηλή.**

- Η ύπαρξη πολλών custom προγραμμάτων προκαλεί πολυπλοκότητα και αύξηση του κόστους και του χρόνου συντήρησης, ειδικά σε upgrades.

#### **Ευκολία προτεινόμενης λύσης.**

- Αρκετά **Δύσκολη**. Απαιτεί συντονισμό και συνεργασία με το Business αλλά και δυναμισμό από τη μεριά του IT. Οι απαιτήσεις των χρηστών θα πρέπει να φιλτράρονται και να υλοποιούνται εφόσον έχουν συγκεντρωθεί όλα τα απαραίτητα στοιχεία.
- Όσον αφορά στο documentation θεωρώ ότι είναι **Μέτριας Δυσκολίας**.
- Η εύρεση χρόνου για ποιοτική διερεύνηση, και επικοινωνητικό πειραματισμό θα πρέπει να είναι το τελικό ζητούμενο του Lean μετασχηματισμού στο SW Development .



### 9.8.6 Ανακεφαλαίωση

Παρακάτω φαίνεται ένας συνοπτικός πίνακας με τα υπάρχοντα προβλήματα, την κατηγοριοποίηση Σπατάλης, τις Lean αρχές που πρέπει να ακολουθηθούν και τις προτεινόμενες λύσεις:

Πρόβλημα	Σπατάλη	Lean αρχές	Προτεινόμενες λύσεις
Απροσδόκητος τερματισμός («Σκάσιμο» Προγ/των)	Ελαττωματικός κώδικας (Defects)	Καθυστέρηση Δέσμευσης/Δημιουργία Ποιότητας	Αύξηση χρόνου Planning /pair programming/ Δημιουργία σεναρίων / OK χρηστών σε όλα τα σενάρια / Επιβεβαίωση καλής λειτουργίας
Διαφορετική εικόνα στα συστήματα Παραγωγικό/DEV/QA	Ημιτελής δουλειά (Partially Done Work)	Ταχύτητα στην υλοποίηση/Δημιουργία Γνώσης	visual boards/15-λεπτο καθημερινό και εβδομαδιαίο meetings/μη παραλαβή νέων incidents
Εξουθένωση συναδέλφων/ κακό εργασιακό κλίμα	Ημιτελής δουλειά (Partially Done Work), Εναλλαγή εργασίας (Task switching)	Optimize the Whole/Σεβασμός στους Ανθρώπους	Διαφορετικές ομάδες υλοποίησης projects-support/ εύρεση του πραγματικού προβλήματος (A3, 5Whys)/ησυχία στο χώρο /εκπαίδευση /coaching/απλοποίηση διαδικασίας εγκρίσεων
Παρενέργειες υλοποιημένων έργων («απόνερα»)	Ημιτελής δουλειά (Partially Done Work)	Καθυστέρηση δέσμευσης/ Δημιουργία Γνώσης/ Εστίαση στην ποιότητα	μεταφορά αλλαγών στην παραγωγή τμηματικά / σενάρια ελέγχου / documentation / εκπαίδευση PMs
Δημιουργία πολλών custom προγραμμάτων	Υπερβάλλουσα λειτουργικότητα, Επανεύρεση λύσης	Optimize the Whole/Καθυστέρηση δέσμευσης/Δημιουργία γνώσης	Περισσότερος χρόνος στη διερεύνηση / Τεκμηρίωση των αλλαγών από τους χρήστες / documentation

Πίνακας 7. Συνοπτικός Πίνακας Προβλημάτων

### 9.9 Ευκολία Υλοποίησης και Χρονοδιάγραμμα Προτεινόμενων Λύσεων.

Είναι προφανές ότι κάποιες από τις προτεινόμενες λύσεις μπορούν να εφαρμοστούν άμεσα, κάποιες απαιτούν υποστήριξη από τη διοίκηση και κάποιες είναι nice to have. Διαπιστώνεται ότι οι περισσότερες βελτιώσεις που προτείνω, αφορούν και στοχεύουν στη βελτίωση της κατάστασης μέσω αλλαγής της κουλτούρας, που απαιτεί συστηματική, συνεχή προσπάθεια και χρόνο.

Στις **Άμεσες** αλλαγές θεωρώ ότι συγκαταλέγονται οι παρακάτω :

- Εκπαίδευση PMs – Προγραμματιστών- Συμβούλων (μπορεί να ενταχθεί στο εκπαιδευτικό πρόγραμμα της εταιρείας)
- Οι κανόνες ονοματολογίας/comments των προγραμμάτων
- Η επιβεβαίωση του χρήστη για τις μεταφερθείσες στην παραγωγή αλλαγές. Οτι δεν υπάρχει δηλ. εκκρεμότητα ή πρόβλημα στο παραγωγικό σύστημα.
- Το καθημερινό 15-λεπτο ενημερωτικό meeting της κάθε ομάδας υποστήριξης.
- Η παρακολούθηση/παρατήρηση της πραγματικής δουλειάς του χρήστη, ώστε να βρεθεί η βέλτιστη λύση.
- Η χρήση του skype/conference calls ώστε να μην ενοχλούνται οι συνάδελφοι από τους θορύβους του open space χώρου.
- Η αύξηση του χρόνου διερεύνησης. Μπορεί να ακούγεται δύσκολο λόγω της πίεσης χρόνου, αλλά είναι ο μόνος τρόπος να δίνονται ορθότερες και ριζικές λύσεις και θα βοηθήσει στη βελτιστοποίηση της όλης κατάστασης.
- Βελτίωση της επικοινωνίας ανάμεσα σε Business- IT και συγκέντρωση όσο γίνεται περισσότερων πληροφοριών για το πρόβλημα.
- Αιτιολόγηση της απαίτησης των χρηστών.
- Δημιουργία documentation με κατάλληλες λέξεις-κλειδιά.

Στις **Μεσοπρόθεσμες** αλλαγές (ορίζοντας τριμήνου για τη λήψη των αποφάσεων και τη θεσμοθέτηση κανόνων και διαδικασιών), για τις οποίες θα χρειαστεί η υποστήριξη της διοίκησης είναι οι παρακάτω :

- Η μη παραλαβή νέων incidents εάν ο χρήστης έχει ήδη δύο σε εξέλιξη.
- Pair programming για βελτίωση και έλεγχο του κώδικα
- Η βελτίωση της ποιότητας των παραδοτέων των projects
- Δημιουργία σεναρίων και ελέγχός τους ως διαδικασία ελέγχου Incidents – Projects.
- Εβδομαδιαίο meeting όλων των ομάδων, για την ενημέρωση των πεπραγμένων.

- Ηλεκτρονική παρακολούθηση των projects (αντί Visual boards)

Οι **Δύσκολες (Nice to have)** είναι οι παρακάτω :

- Visual Boards για άμεση ενημέρωση του status των projects.
- Ο διαχωρισμός των πόρων των projects – support.
- Η αλλαγή του open space.
- Η άμεση εμπλοκή της ανώτερης διοίκησης στην καθημερινότητα του IT (αν και θεωρώ ότι είναι υψίστης σημασίας)
- Η αλλαγή κουλτούρας όσον αφορά το scope των projects. Συνήθως είναι κλειστά και μάλλον έτσι θα συνεχίσουν.
- Η τμηματοποίηση της υλοποίησης είναι ένα άλλο ζήτημα το οποίο θα αντιμετωπιστεί με σκεπτικισμό.
- Η χρήση του A3 ως μέσο σκέψης και επίλυσης προβλημάτων.
- Το coaching, διότι απαιτεί αλλαγή κουλτούρας.

### 9.10 KPIs για την Υποστήριξη της εφαρμογής του Lean

Η όλη προσπάθεια για την εφαρμογή του Lean, η διοίκηση του project χρειάζεται να υποστηριχθεί μέσω εργαλείων που θα παρακολουθούν την πρόοδο που έχει επιτευχθεί, όσο και για την χάραξη προτεραιοτήτων. Ένα πολύ χρήσιμο σύστημα είναι η εφαρμογή Δεικτών Απόδοσης (KPIs)

Τόσο οι μεσοπρόθεσμες όσο και οι δύσκολες προτεινόμενες αλλαγές, θα είναι ευκολότερο να υποστηριχτούν εάν οι άμεσες αλλαγές εμφανίσουν κάποιο μετρήσιμο αποτέλεσμα. Για να υπάρχει όμως σύγκριση (πριν-μετά), θα πρέπει να υπάρχουν αξιόπιστες μετρήσεις πριν ξεκινήσει το project, και συνεπώς ο σχεδιασμός και εφαρμογή συστήματος KPIs έχει άμεση προτεραιότητα.

Παραθέτω τους βασικούς δείκτες που κρίνω χρήσιμους στην παρούσα φάση:

#### 1. Τήρηση SLA(Service Level Agreement)

Απαραίτητη μέτρηση αφορά στην απόδοση της επίλυσης των προβλημάτων. Δηλ. εάν βάσει της SLA(Service Level Agreement), τα προβλήματα αντιμετωπίζονται μέσα στον προβλεπόμενο χρόνο. Η μέτρηση ουσιαστικά αφορά το χρόνο κύκλου των value streams. Τα ποσοστά που θεωρούνται ικανοποιητικά πρέπει να

αποφασίζονται σε συνεργασία με το Business. (Π.χ. για τα High Urgency Incidents που πρέπει να αντιμετωπιστούν εντός τριών ωρών, εάν ο δείκτης είναι 95% θεωρούμε ότι είμαστε σε καλό δρόμο.)

2. Ποσοστό High Urgency Incidents

Ένας άλλος δείκτης θα μπορούσε να είναι η συχνότητα των High Urgency Incidents. Εάν συμβαίνουν συχνά τέτοιου είδους incidents, σίγουρα χρειάζεται ανάλυση των αιτιών και εύρεση ριζικής λύσης για την αντιμετώπισή τους.

3. Επανάληψη Incidents

Άλλη μέτρηση θα πρέπει να αφορά γενικότερα στην επανάληψη ίδιων incidents. Εάν δηλ. έρχονται συχνά αλλαγές για το ίδιο πρόγραμμα/transaction/functionality, αυτό εμπίπτει σε ανάλυση των αιτιών και εύρεση ριζικής λύσης για την αντιμετώπισή τους.

4. User Action incidents σε Αναμονή

Άλλη βασική και απλή μέτρηση είναι τα User Action incidents, που περιμένουν UAT/απάντηση από το χρήστη. Προτείνω τη μη λήψη νέων incidents από το συγκεκριμένο χρήστη, εάν δεν ολοκληρώσει εκείνα που χρωστάει.

5. Βαθμός Ικανοποίησης Πελατών

Σημαντική μέτρηση αφορά την ικανοποίηση των πελατών/χρηστών, με βαθμολόγηση π.χ. των incidents/projects και εάν όντως αντιμετώπισαν επιτυχώς τα συγκεκριμένα προβλήματα που προσπαθούσαν να επιλύσουν.

6. Βαθμός Ικανοποίησης Προσωπικού IT

Εξίσου σημαντική όμως είναι και η μέτρηση της ικανοποίησης των συναδέλφων στο IT. Στην εβδομαδιαία συνάντηση ή σε κάποια προκαθορισμένη για αυτόν το σκοπό, θα μπορούσαν να συζητούνται τρόποι βελτίωσης των διαδικασιών και γενικότερα του εργασιακού περιβάλλοντος.

### 9.11 Τελικά συμπεράσματα.

Θεωρώ ότι με την επιτυχή εφαρμογή των παραπάνω προτάσεων (Kaizen project, καθημερινό Kaizen, εξάλειψη σπατάλης), το συγκεκριμένο Τμήμα θα βελτιώσει κατά πολύ τόσο την εικόνα του προς τα έξω όσο και τη ζωή των εργαζομένων του. Θα έχουν περισσότερο χρόνο για πειραματισμό και καινοτομία, αντί του να ασχολούνται με τα ίδια συνέχεια πράγματα. Θα προσδώσουν προστιθέμενη αξία στους πελάτες/χρήστες αλλά θα νιώθουν και οι ίδιοι υπερηφάνεια και ικανοποίηση για τη δουλειά τους.

Όπως έχει τονιστεί η επιτυχία ενός project για την εφαρμογή του Lean στο IT, προϋποθέτει την υποστήριξη του management. Για να επιτευχθούν τα προσδοκώμενα αποτελέσματα θα πρέπει και η διοίκηση του Τμήματος να είναι αρωγός και υποστηρικτής, όπως και η διοίκηση της εταιρείας

Η επιτυχία του project που προτείνω, μπορεί να αποτελέσει τη βάση ενός πολύ μεγαλύτερου project, με την υποστήριξη εξειδικευμένων συμβούλων, για τον Lean μετασχηματισμό όχι μόνο του τμήματος IT αλλά και ολόκληρης της εταιρείας, καθώς είναι δύσκολο να επιβιώσει μια κουλτούρα Lean ως «νησίδα» μέσα σε ένα οργανισμό όπου είναι κυρίαρχη μια άλλη κουλτούρα.

## ΠΑΡΑΡΤΗΜΑ Ι

### Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

© 2001, the above authors  
this declaration may be freely copied in any form,  
but only in its entirety through this notice.

### **Principles behind the Agile Manifesto**

*We follow these principles:*

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

**ΠΑΡΑΡΤΗΜΑ II****Σπατάλες στο IT και στο SW Development**

Παρατίθεται λίστα με σπατάλες στο χώρο του IT, χωρίς αυτό να σημαίνει ότι σε κάθε εταιρεία δεν μπορεί να διαγνωστούν και άλλες.

Environmental Waste	Business processes causing environmental waste, that could be improved with thoughtful application and information systems
Environmental Waste	Document storage, archive, and disposal waste
Environmental Waste	Electricity consumed by IT assets
Environmental Waste	Improper disposal of materials, including hazardous materials found in many electronics
Environmental Waste	Ink, toner cartridges, and other recyclables that are disposed prematurely ... or not recycled!
Environmental Waste	Overnight shipping expenses and carbon footprint that could be reduced through electronic transport
Environmental Waste	Paper consumed by unnecessary printing
Environmental Waste	Printing single-sided versus double-sided
Environmental Waste	Transportation (fuel, carbon, and other costs) that could be reduced through telecommuting and remote meetings
Environmental Waste	Using paper coffee cups instead of ceramic mugs
Errors/Defects/Rework	Application bugs and design flaws
Errors/Defects/Rework	Assuming customer needs / requirements instead of getting it directly from the customer (not asking for information, or asking for the wrong information)
Errors/Defects/Rework	Consequences of estimating, pricing, forecasting, or scheduling errors (to name just a few) based on incomplete, incorrect, irrelevant or obsolete data or assumptions
Errors/Defects/Rework	E-mail, spam, and the problems that result from excessive filtering
Errors/Defects/Rework	Entering conflicting information into the same or multiple systems
Errors/Defects/Rework	Errors resulting from insufficient or improper analysis and reflection
Errors/Defects/Rework	Excessive productivity standards, measurements, and incentives that emphasize volume/speed over quality
Errors/Defects/Rework	Forming a kaizen team to improve an automated business process without including a representative from IT, often resulting in incorrect system requirements and change requests
Errors/Defects/Rework	Free text fields instead of drop-downs, check boxes, etc. that allow user error and propagation of bad data throughout downstream systems
Errors/Defects/Rework	Helpdesk interventions that do not solve the problem due to misunderstanding, unclear communication or problem definition thus requiring further troubleshooting and correction
Errors/Defects/Rework	Helpdesk knowledgebase information that is incorrect, incomplete, unclear, or obsolete resulting in a failed intervention, potentially causing additional harm and lost user productivity
Errors/Defects/Rework	Inability to receive email attachments that exceed file size limits, causing rework
Errors/Defects/Rework	Inadequately tested changes to software and systems
Errors/Defects/Rework	Incomplete, incorrect, obsolete, or unclear information from upstream processes that must be corrected
Errors/Defects/Rework	Inspection and correction processes



Errors/Defects/Re work	Lost information
Errors/Defects/Re work	Measures that lack valid prescriptive guidance
Errors/Defects/Re work	Meeting conflicts and subsequent rescheduling caused by misuse of the scheduling system
Errors/Defects/Re work	Missing, inaccurate, incomplete, unclear, or obsolete specifications or process documentation
Errors/Defects/Re work	Mixed messages from management about the importance of doing work right the first time, while also doing all daily work "on schedule"
Errors/Defects/Re work	Not understanding customer/user requirements and expectations, thus delivering ineffective "solutions"
Errors/Defects/Re work	Poor project execution
Errors/Defects/Re work	Reconciling conflicting sources of data and subsequent disagreements over whose data is right
Errors/Defects/Re work	Requiring workers to memorize or write down data from one source to be entered into another source
Errors/Defects/Re work	Software patches applied simply to correct errors introduced by the previous patches
Errors/Defects/Re work	Testing introduced too late in the lifecycle of a project
Errors/Defects/Re work	Time and effort required to identify and correct bad data as it propagates across multiple data sources
Errors/Defects/Re work	Unauthorized changes to software and systems
Errors/Defects/Re work	Unstable software, hardware, communications, devices, etc., that make it difficult for people to get value-added work done
Inventory	Backlog accumulating in the software development, support, or enhancement queue
Inventory	Backlog accumulating in electronic and physical inboxes
Inventory	Clutter: physical and virtual disorganization wherever work is done or knowledge is stored and shared
Inventory	Endless spreadsheets and other "productivity enhancing" documents that encourage local optimization while fragmenting the overall process, increasing the number of delays, handoffs, and errors
Inventory	Excess information across local drives, shared drives, SharePoint sites, datawarehouses, duplication of the same data in multiple forms (e.g. Electronic and paper)
Inventory	Excessive inventory of ink cartridges and other consumables that require space and may become damaged before they are needed
Inventory	Excessive parts inventory
Inventory	Introducing excessive tools, technologies, or methodologies
Inventory	Multiple software code objects that perform the same function
Inventory	Old or obsolete electronic and physical files and messages that should be stored, disposed, or recycled
Inventory	Partially completed development work—uncoded documentation, unsynchronized code, untested code, undocumented code, undeployed code
Inventory	Projects and programs that are not being considered but are still on the backlog

Inventory	Sending attachments rather than links, which creates multiple versions of a document (also causing version control and security problems)
Inventory	Software that's purchased but never deployed
Inventory	Unnecessary, unclear, incorrect, obsolete, or unused documentation
Inventory	Unused/unnecessary software user licenses
Inventory	Work waiting to be reviewed, approved, and forwarded

Motion/Transport/ Transfer	E-mails sent/responded to uninvolved people
Motion/Transport/ Transfer	Filing or moving physical documents that could be stored and forwarded or linked electronically
Motion/Transport/ Transfer	Intercompany handoffs of nonstandard, nonintegrated information (e.g. supply chain transactions)
Motion/Transport/ Transfer	Manual expediting of documents—walking documents through the approval process
Motion/Transport/ Transfer	Needing to split email attachments into smaller segments due to file size limitations
Motion/Transport/ Transfer	Nonstandard handoff of information from one process step to another when it requires some form of change or re-interpretation of the information
Motion/Transport/ Transfer	On-site visits to resolve system issues that could be resolved (or prevented) with remote monitoring and correction
Motion/Transport/ Transfer	Poor ergonomic design of work stations
Motion/Transport/ Transfer	Poor user interface or process design that causes unnecessary keystrokes, mouse clicks, or navigation steps
Motion/Transport/ Transfer	Printing information from one computer then re-entering it into another
Motion/Transport/ Transfer	Providing a spreadsheet or report to more people than actually need it
Motion/Transport/ Transfer	Security barriers to flow of nonsensitive information
Motion/Transport/ Transfer	Sending attachments rather than links to documents
Motion/Transport/ Transfer	Toggling between disconnected applications
Motion/Transport/ Transfer	Unnecessary movement of electronic information
Motion/Transport/ Transfer	Using multiple emails for dialogue when a conference call or face-to-face meeting is more effective
Motion/Transport/ Transfer	Walking to/from printer, copier, fax machine, filing cabinet, archival storage
Overprocessing	“Gold plating” system design and performance striving for perfection rather than practical adequacy
Overprocessing	Budgeting tasks that do not lead to better decision making
Overprocessing	Capturing data at various points along a process, rather than capturing at the source
Overprocessing	Cost accounting efforts that track and calculate cost per unit/order/event rather than the value stream cost
Overprocessing	Entering redundant information into the same or multiple systems
Overprocessing	Excessive analysis that does not add value to a decision
Overprocessing	Excessive automated exception notifications and alerts that cause distraction, but do not drive effective action or decision making
Overprocessing	Excessive or unused reason or categorization codes configured into the software
Overprocessing	Excessive, unnecessarily invasive, or complex security policies and procedures
Overprocessing	Failure to cancel a project when it is determined to be no longer viable
Overprocessing	Frequent switching between incomplete jobs, also known as multitasking, thrashing, task switching
Overprocessing	Gridlocked or unclear matrix management practices
Overprocessing	Helpdesk troubleshooting that addresses symptoms but not root causes
Overprocessing	Ineffective or repetitive meetings that add little value and don't resolve lingering issues
Overprocessing	Information addiction, excessive e-mail, web surfing, social networking, and so on..
Overprocessing	Inspection and correction activities required to catch and correct errors that should be prevented by building quality into the process

Overprocessing	Inviting people to unnecessary meetings, or meetings where they don't need to be involved
Overprocessing	Lack of system and process integration that causes unnecessary work
Overprocessing	Management reporting systems (e.g., hierarchal scorecards and dashboards) that do not align strategy with daily activity, and thus do not effectively prioritize action, leading to effort that does not contribute to organization goals (doing the wrong work, but doing it well)
Overprocessing	Mandated but unnecessary or overly complex processes and activities
Overprocessing	Multiple reports containing the same or similar data
Overprocessing	Non-value-adding control activities solely for compliance purposes
Overprocessing	Printed information where electronic is sufficient, for example, presentation slides and handouts
Overprocessing	Printing information from one computer system then re-entering it into another
Overprocessing	Producing and distributing reports that contain information which is not used
Overprocessing	Searching for information or materials that are difficult to find
Overprocessing	Sending people unnecessary messages, such as selecting "reply all" to an e-mail message, when "reply" (or no response) is sufficient
Overprocessing	Trying to govern or manage systems or processes that are unreliable, unstable, undocumented, or too complicated to understand
Overprocessing	Unclear lines of decision making responsibility/authority that cause unnecessary changes, time, cost, errors, and rework once a project is underway
Overprocessing	Unnecessary monthly closing activities
Overprocessing	Unnecessary transactions, capturing more information than is needed to complete the process or to support decision making (e.g. capturing standard cost variances and calculating overhead allocations in order to support a process that flows)
Overprocessing	Unused functionality designed into software
Overprocessing	Workflow approval routings that are not necessary
Overprocessing	Writing something, then entering it into a computer system
Overproduction	Making and locking decisions in too early that commit resources and limit flexibility
Overproduction	Making extra copies
Overproduction	Printing collateral materials that are quickly obsolete
Overproduction	Printing single-sided versus double-sided
Overproduction	Processing paperwork or information before the downstream operator is ready for it
Overproduction	Producing and distributing a document days before a meeting, that is modified based upon later information and redistributed
Overproduction	Software requirements that are specified long before coding
Overproduction	Unnecessary or early work performed due to unclear priorities, interdependencies, or sequence
Underutilized Human Potential	Dull, repetitive, or mundane activities that could be eliminated or automated
Underutilized Human Potential	Excessive approvals needed or other bureaucracy that discourages improvement and innovation
Underutilized Human Potential	Excessively detailed standards that do not allow flexibility
Underutilized Human Potential	Information overload
Underutilized Human Potential	Lack of guidelines or work standards to empower employees
Underutilized Human Potential	Lack of visuals and measures to indicate where problems need to be solved
Underutilized Human Potential	Lost, conflicting, or underutilized knowledge that must be rediscovered and relearned once a problem occurs
Underutilized Human Potential	Not asking for ideas

Underutilized Human Potential	Not capturing and sharing ideas
Underutilized Human Potential	Not communicating targeted objectives to all associates involved in a process
Underutilized Human Potential	Not encouraging cross-functional communication and collaboration
Underutilized Human Potential	Not investing in education and training
Underutilized Human Potential	Not making knowledge easy to locate and apply

## **ΒΙΒΛΙΟΓΡΑΦΙΑ**

Akhilesh N Singh, Avinash Singh (2018). Lean IT Principles to Practice. Notion Press

Ballard, Glenn. "Positive vs. Negative Iteration in Design." Proceedings Eighth Annual Conference of the International Group for Lean Construction, IGLC-6, Brighton, UK, July 17–19, 2000.

Bell Steven- Orzen Mike, Lean IT-Enabling and Sustaining your Lean Transformation", 2010

Bowen, D. E., & Youngdahl, W. E. (1998). Lean service: in defense of a production-line approach. *International Journal of Service Industry Management* , 9 (3), 207-225.  
<http://dx.doi.org/10.1108/09564239810223510> (τελευταία πρόσβαση 06/01/2020)

Deming Edwards, Out of the Crisis, MIT Press, 2000. Originally published 1982

George Michael and Wilson Stephen, Conquering Complexity in Your Business: How Wal-Mart, Toyota, and Other Top Companies Are Breaking Through the Ceiling on Profits and Growth, McGraw-Hill, 2004.

Hamel Gary, Management Innovation, Harvard Business Review, 02.2006

Higor Leite, 2015

[https://www.researchgate.net/publication/273912843\\_Lean\\_philosophy\\_and\\_its\\_applications\\_in\\_the\\_service\\_industry\\_A\\_review\\_of\\_the\\_current\\_knowledge](https://www.researchgate.net/publication/273912843_Lean_philosophy_and_its_applications_in_the_service_industry_A_review_of_the_current_knowledge) (τελευταία πρόσβαση 06/01/2020)

MacCormack Alan, "Product-Development Practices That Work: How Internet Companies Build Software", MIT Sloan Management Review, Winter 2001, Vol 40 number 2.

MacGibbon P. Simon, Schumacher R. Jeffrey, and Tinaikar S. Ranjit, When IT's Customers Are External, McKinsey Quarterly, Q1 2006

Maister, D. H. (1985). The psychology of waiting lines. In J. A. Czepiel, M. R. Solomon & C. F. Suprenant (Eds.), *The service Encounter: Managing Employee/Customer Interaction in service Business*. Lexington: Lexington Books. PMID:4037768

Minoura Teruyuki:

[https://media.toyota.co.uk/wp-content/files\\_mf/1323862732essenceTPS.pdf](https://media.toyota.co.uk/wp-content/files_mf/1323862732essenceTPS.pdf), (τελευταία πρόσβαση 21/11/2019)

Morgan James and Liker Jeffrey, *The Toyota Product Development System: Integrating People, Process, and Technology*, Productivity Press, 2006.

Ohno Taiichi, "Toyota Production System: Beyond Large Scale Production", Productivity Press, 1988

Athens MBA, Εφαρμογή της Lean Φιλοσοφίας σε εταιρικό Τμήμα Ανάπτυξης Λογισμικού

Poppendieck Mary and Tom, Implementing Lean Software Development , From Concept to Cash, Addison-Wesley 2007.

Poppendieck Mary and Tom, Lean Software Development: An Agile Toolkit, Addison-Wesley 2003

Reichheld Fred, The Ultimate Question:Driving Good Profits and True Growth, Harvard Business School Press, 2006

Senge Peter, The Fifth Discipline, Doubleday, 1990

Shingo Shigeo, Study of “Toyoda” Production System from an Industrial Engineering Viewpoint, Productivity Press,1981.

Vasilash S . Gary, “Engaging the ES 300”, Automotive Design and Production, September,2001.

Wiering Robert, [https://www.academia.edu/31561484/Lean-IT\\_Waste\\_Checklist](https://www.academia.edu/31561484/Lean-IT_Waste_Checklist) (τελευταία πρόσβαση 14/01/2020)

Womack, J. P., & Jones, D. T. (2003), Lean thinking, Simon & Schuster

Womack, J. P., & Jones, D. T. (2005). Lean Consumption. Harvard Business Review

<https://agilemanifesto.org/> (τελευταία πρόσβαση 14/01/2020)

<https://www.allaboutlean.com/230-years-interchangeability/> (τελευταία πρόσβαση 14/01/2020)

<https://amplifiedgrowth.net/2017/02/03/associations-can-learn-alfred-sloan-not-henry-ford/> (τελευταία πρόσβαση 14/01/2020)

<https://www.britannica.com/biography/Eli-Whitney> (τελευταία πρόσβαση 14/01/2020)

<https://www.coursera.org/lecture/lean-manufacturing-services/the-four-key-differences-between-lean-in-manufacturing-and-in-services-rG0jL> (τελευταία πρόσβαση 03/02/2020)

<https://www.infoq.com/articles/standish-chaos-2015/> (τελευταία πρόσβαση 06/01/2020)

<https://www.toolshero.com/toolsheroes/william-edwards-deming/> (τελευταία πρόσβαση 14/01/2020)