



ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ
ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΑ
ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ (M.I.S.)

**Στρατηγικές νέων προϊόντων λογισμικού/
New Software product strategies**

Μάθημα/ Course:

Στρατηγικές για Πληροφορική και Τηλεματική /
Strategies for Computer Networks and Telematic
Applications_

Καθηγητής/ Professor:

Αναστάσιος Α. Οικονομίδης/
Anastasios A. Economides

Συγγραφέας/ Author:

Άγγελος Γεωργούλας/
Angelos Georgoulas

Θεσσαλονίκη
Μάιος, 2011

Περιεχόμενα

<u>Περίληψη.....</u>	<u>4</u>
<u>Abstract.....</u>	<u>4</u>
<u>Εισαγωγή.....</u>	<u>5</u>
<u>Πληροφορίες για την δημιουργία ενός νέου προϊόντος.....</u>	<u>5</u>
<u>Διαφορές προϊόντων λογισμικού και υπηρεσιών με τα υπόλοιπα προϊόντα.....</u>	<u>6</u>
<u>Κατηγορίες εταιριών παραγωγής λογισμικού.....</u>	<u>6</u>
<u>Εξειδίκευση ή τυποποίηση στις εταιρίες παραγωγής λογισμικού.....</u>	<u>7</u>
<u>Ανάγκη για εξειδικευμένο και πρωτότυπο λογισμικό.....</u>	<u>7</u>
<u>Δυσκολίες στην παραγωγή λογισμικού.....</u>	<u>8</u>
<u>Στρατηγικές που ακολουθούν οι επιχειρήσεις ανάλογα με την θέση τους στον κλάδο...9</u>	
<u>Στρατηγικές αποφάσεις που ακολουθούν μικρές εταιρίες παραγωγής λογισμικού.....</u>	<u>10</u>
<u>Βέλτιστος χρόνος για την έκδοση νέου λογισμικού.....</u>	<u>11</u>
<u>Αξιολόγηση λογισμικού.....</u>	<u>13</u>
<u>Συμπεράσματα.....</u>	<u>14</u>
<u>Προτάσεις για μελλοντική έρευνα.....</u>	<u>14</u>
<u>Βιβλιογραφία.....</u>	<u>15</u>

Table of Contents

Περίληψη.....	4
Abstract.....	4
Εισαγωγή.....	5
Πληροφορίες για την δημιουργία ενός νέου προϊόντος.....	5
Διαφορές προϊόντων λογισμικού και υπηρεσιών με τα υπόλοιπα προϊόντα.....	6
Κατηγορίες εταιριών παραγωγής λογισμικού.....	6
Εξειδίκευση ή τυποποίηση στις εταιρίες παραγωγής λογισμικού.....	7
Ανάγκη για εξειδικευμένο και πρωτότυπο λογισμικό.....	7
Δυσκολίες στην παραγωγή λογισμικού.....	8
Στρατηγικές που ακολουθούν οι επιχειρήσεις ανάλογα με την θέση τους στον κλάδο...9	
Στρατηγικές αποφάσεις που ακολουθούν μικρές εταιρίες παραγωγής λογισμικού.....	10
Βέλτιστος χρόνος για την έκδοση νέου λογισμικού.....	11
Αξιολόγηση λογισμικού.....	13
Συμπεράσματα.....	14
Προτάσεις για μελλοντική έρευνα.....	14
Βιβλιογραφία.....	15

Περίληψη

Η εργασία πραγματεύεται τις στρατηγικές που πρέπει να ακολουθήσει μία επιχείρηση κατασκευής προϊόντων λογισμικού. Στα πλαίσια της προσπάθειας να βρεθεί η καλύτερη στρατηγική, θα παρατεθούν οι διαφορές των προϊόντων λογισμικού με τα υπόλοιπα προϊόντα, καθώς και οι ανάγκες της αγοράς, αφού πρώτα συλλεχθούν όλες οι πληροφορίες που είναι απαραίτητες για την παραγωγή των προϊόντων αυτών. Θα αναφερθούν τόσο ο ιδανικός χρόνος για την κατασκευή του λογισμικού, όσο και οι δυσκολίες στην παραγωγή από την πλευρά του μάνατζερ, αλλά και του προγραμματιστή. Αναφορικά με τις εταιρίες, θα μελετηθούν ανάλογα με την θέση τους στον κλάδο, τις βλέψεις τους και τις σχέσεις ανταγωνισμού. Στο τέλος της εργασίας θα αναφερθούν αφενός ο βέλτιστος χρόνος για την έκδοση ενός νέου λογισμικού και αφετέρου τα κριτήρια με τα οποία αξιολογείται ένα προϊόν λογισμικού.

Abstract

In this paper it is discussed the topic of what strategies a company which produces software products must follow. In an effort to find out the best strategy, will be set out the differences in software products with other products as well as the market needs, and all these will be done after collecting all those information which are necessary for the production of these products. Besides, it will be reported the ideal time for the production of software and also the difficulties in the production as far as the side of the manager and of the programmer. As far as the companies are concerned, they will be studied according to their situation in the industry, their aspirations and the competition. In the end it will be reported the optimal time for the launch of a new software and with what criteria a software product is assessed.

Εισαγωγή

Ως ανάπτυξη λογισμικού ορίζεται η κατασκευή ενός προϊόντος λογισμικού, που είναι γνωστή και ως ανάπτυξη εφαρμογής, σχεδιασμός λογισμικού, ανάπτυξη εφαρμογής λογισμικού ή ανάπτυξη πλατφόρμας. Ο όρος ανάπτυξη λογισμικού μπορεί να αναφερθεί και στην δραστηριότητα του προγραμματισμού ενός υπολογιστή, που είναι η διαδικασία της εγγραφής και διατήρησης του πηγαίου κώδικα. Με την ευρύτερη έννοια του όρου, πρόκειται για τη διαδικασία που ξεκινά με τη σύλληψη της ιδέας και τελειώνει με την τελική υλοποίηση του λογισμικού, μέσω μίας δομημένης και προγραμματισμένης διεργασίας. Έτσι, η ανάπτυξη λογισμικού μπορεί να εμπεριέχει την έρευνα, την νέα ανάπτυξη, την πρωτοτυπία, την αναδιοργάνωση, την επαναχρησιμοποίηση, την αναδιοργάνωση, την διατήρηση του λογισμικού ή όποιες άλλες δραστηριότητες συντελούν στην κατασκευή του.

Οι τρεις πιο πιθανοί λόγοι για την ανάπτυξη ενός λογισμικού είναι η κάλυψη των ειδικών αναγκών ενός συγκεκριμένου πελάτη/ επιχείρησης, η κάλυψη μίας αντιληπτής ανάγκης μιας ομάδας πιθανών χρηστών είτε η προσωπική χρήση.

Η ανάγκη για καλύτερο έλεγχο ποιότητας της διαδικασίας ανάπτυξης λογισμικού επέφερε ως αποτέλεσμα πιο αυστηρούς ελέγχους, που στοχεύουν στην εφαρμογή της συστημικής προσέγγισης στην παραγωγή λογισμικού (http://en.wikipedia.org/wiki/Software_development).

Πληροφορίες για την δημιουργία ενός νέου προϊόντος

Η επιτυχία ενός νέου προϊόντος υψηλής τεχνολογίας εξαρτάται κυρίως από την καινοτομία και από την τεχνολογία που χρησιμοποιεί. Επιπλέον, μεγάλη σημασία έχουν οι πληροφορίες που συλλέγονται από την αγορά για τις ανάγκες που υπάρχουν. Σχετικό παράδειγμα αποτελεί η Sony, όπου οι υπεύθυνοι ήταν αρχικά επιφυλακτικοί για τη χρησιμοποίηση τέτοιων πληροφοριών, όμως η επιτυχία του υπολογιστή Vaio-W, απέδειξε τη λανθασμένη προσέγγισή τους. Η Sony ανέπτυξε το Vaio-W επειδή έρευνα που έκανε πριν την δημιουργία του έδειξε πως οι Ιάπωνες ήθελαν φορητούς υπολογιστές λόγω του μικρού τους μεγέθους. Το αποτέλεσμα ήταν το συγκεκριμένο μοντέλο της Sony να πωληθεί την πρώτη μέρα παραπάνω από όσο ήλπιζαν για ολόκληρο τον πρώτο μήνα κυκλοφορίας του (Velhuizen, Hultink and Griffin, 2006).

Σύμφωνα με την Μοτορόλα, η ομάδα δημιουργίας του νέου προϊόντος πρέπει να παραδώσει στην διοίκηση τις εξής πληροφορίες (Zangwill, 1993, σελ.206-207):

1. Στρατηγική Μάρκετινγκ

- Ανάγκες των πελατών
- Αποτελέσματα έρευνα αγοράς
- Παράθυρα ευκαιρίας
- Αναγνώριση ρίσκου αγοράς και εμποδίων
- Στρατηγική διανομής/ απαιτήσεις για την διανομή
- Προβλέψεις πωλήσεων μέσω market-by-market, ιστορία πωλήσεων, τιμολόγησης βάση τιμής αγοράς, στρατηγική παροχής υπηρεσιών, ανταγωνισμός και άλλοι εξωγενείς παράγοντες

2. Περιγραφή της παραγωγής

- Βασικά χαρακτηριστικά/ ιδιότητες/ προσδοκώμενο κόστος παραγωγής
- Απαιτούμενη και διαθέσιμη τεχνολογία
- Αξιολόγηση παραγωγικής διαδικασίας
- Αναγνώριση κινδύνου και αξιολόγηση τεχνολογίας

3. Προϋπολογισμός δαπάνης της επένδυσης

- Προτεινόμενο χρονοδιάγραμμα
- Πειθαρχημένη εκτίμηση αναγκαίων πόρων
- Εκτιμώμενο συνολικό κόστος του έργου

4. Χρηματοοικονομικές προβλέψεις

- Απαραίτητη επένδυση, υπολογισμός καθαρής παρούσας αξίας, και εσωτερικό ποσοστό απόδοσης
- Πρόβλεψη κερδών

5. Σύνοψη και συμπεράσματα

- Προβλεπόμενο χρονοδιάγραμμα για την επόμενη φάση της ανάπτυξης
- Προβλεπόμενοι πόροι για την επόμενη φάση της ανάπτυξης
- Προτεινόμενες ενέργειες

Διαφορές προϊόντων λογισμικού και υπηρεσιών με τα υπόλοιπα προϊόντα

Ένα κοινό προϊόν έχει ως κύκλο ζωής πέντε φάσεις: α) την εισαγωγή του στην αγορά, β) την ανάπτυξή του μέσω της ζήτησης και της πώλησης στην αγορά, γ) την ωρίμανση του, δ) την παρακμή του και ε) το θάνατό του. Ωστόσο, στα προϊόντα λογισμικού υψηλής τεχνολογίας, όπως και στις υπηρεσίες, ο κύκλος ζωής περιορίζεται από την στιγμή που δίνεται προς κατανάλωση ως την στιγμή που αναβαθμίζεται και ξαναδίνεται για κατανάλωση στους πελάτες. Αυτό που διαφοροποιεί τα προϊόντα λογισμικού από τα υπόλοιπα προϊόντα δεν είναι η τεχνολογία με την οποία κατασκευάζονται αλλά η τεχνολογία που διαθέτουν μέσα τους. Στα προϊόντα υψηλής τεχνολογίας λογισμικού αντί να ανακαλύπτεται συνεχώς ο τρόπος για να παραχθεί το νέο προϊόν, ο δημιουργός τους τα κεφαλαιοποιεί με βάση τις ομοιοτητές τους, προκειμένου να δημιουργήσει νέες εφαρμογές χρησιμοποιώντας χειροποίητα αντικείμενα λογισμικού και κομμάτια. Έτσι τα προϊόντα λογισμικού πρέπει να έχουν επεκτασιμότητα, δυνατότητα επαναχρησιμοποίησης και προσαρμοστικότητα στα συστήματα (Aramand, 2008).

Δηλαδή, για να εξασφαλιστεί η συνεχής ποιότητα ενός λογισμικού, νέες του εκδόσεις είναι απαραίτητες. Αυτές οι νέες εκδόσεις προσφέρουν στον καταναλωτή βελτιωμένες και χωρίς προβλήματα υπηρεσίες και η διαδικασία αυτή συνεχίζεται όσο υπάρχει ο κύκλος ζωής του συγκεκριμένου είδους λογισμικού (Levin and Yadid, 1990).

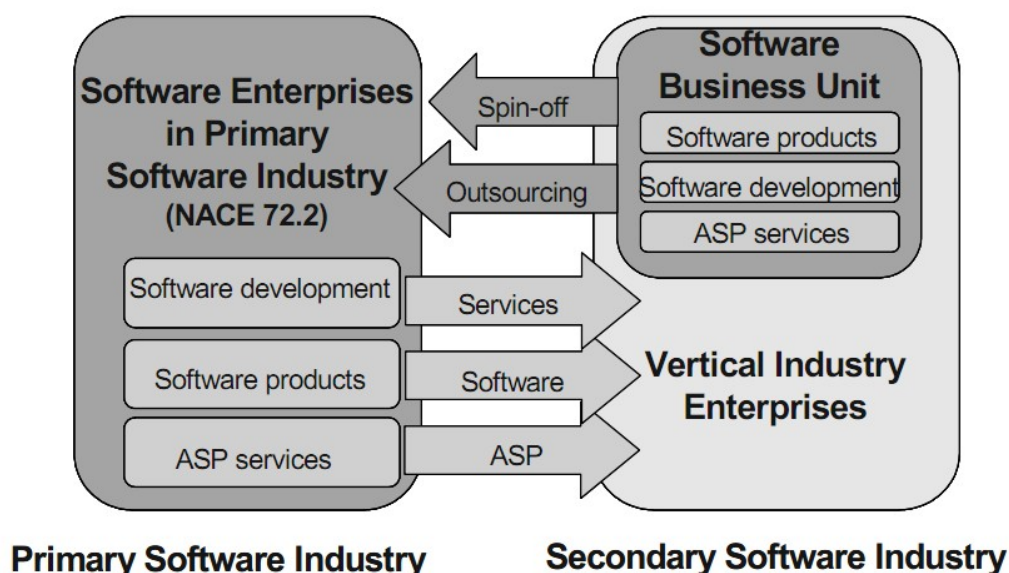
Κατηγορίες εταιριών παραγωγής λογισμικού

Οι απαρχές της βιομηχανίας παραγωγής λογισμικού εντοπίζονται στην απόφαση της IBM να διαχωρίσει μερικά δευτερεύοντα κομμάτια από την ανάπτυξη υπολογιστών σε ανεξάρτητες εταιρίες λογισμικού. Τα πρώτα χρόνια η παραγωγή λογισμικού ήταν μία δευτερεύουσα δραστηριότητα των εταιριών παραγωγής

hardware, ωστόσο κάποιες εταιρίες όπως η IBM συνεχίζουν ακόμη και σήμερα να παράγουν το hardware και το λογισμικό που χρησιμοποιούν. Σταδιακά, η βιομηχανία παραγωγής λογισμικού καθιερώθηκε σαν μια ξεχωριστή βιομηχανία. Η έρευνα για την παραγωγή λογισμικού είχε ως στόχο πρώτα τον πυρήνα της ανερχόμενης βιομηχανίας, δηλαδή τις εταιρίες που ανέπτυσαν και πουλούσαν ανεξάρτητα προϊόντα λογισμικού και υπηρεσίες. Αυτές οι εταιρίες αναφέρονται συχνά ως πρωταρχική βιομηχανία λογισμικού. Ο όρος δευτερεύουσα βιομηχανία λογισμικού αναφέρεται στις επιχειρήσεις λογισμικού που εστιάζουν σε κάποια άλλη δραστηριότητα, αλλά χρησιμοποιούν το λογισμικό ως κομμάτι των προϊόντων ή των υπηρεσιών τους. Οι εταιρίες τηλεπικοινωνιών και παραγωγής αυτοκινήτων είναι δύο παραδείγματα δευτερεύουσας βιομηχανίας λογισμικού (Tyrvaenen, Warsta and Seppanen, 2008).

Εικόνα 1.

Αλληλεπίδραση μεταξύ παραγωγών και καταναλωτών λογισμικού(Tyrvaenen, Warsta and Seppanen, 2008).



Στην εικόνα φαίνεται η αλληλεπίδραση μεταξύ επιχειρήσεων της βιομηχανίας παραγωγής λογισμικού και επιχειρήσεων που τα χρησιμοποιούν.

Εξειδίκευση ή τυποποίηση στις εταιρίες παραγωγής λογισμικού

Είναι σύνηθες φαινόμενο οι εταιρίες που παράγουν νέα προϊόντα να έχουν περιορισμένη προσοχή στην στρατηγική για την διαδικασία παραγωγής νέων προϊόντων, πράγμα που προκαλεί έκπληξη δεδομένου ότι η διαδικασία αυτή και κατ' επέκταση το νέο προϊόν είναι στρατηγικής σημασίας για την επιχείρηση και έτσι είναι πολύ πιθανό να επηρεαστεί από τις στρατηγικές επιλογές της επιχείρησης (Frambach, Prabhu and Verhallen, 2003).

Οι εταιρίες παραγωγής λογισμικού έχουν μεγάλη ποικιλία στρατηγικών για την παραγωγή προϊόντων, με τις οποίες προσπαθούν να ανταποκριθούν στις ανάγκες των πελατών μέσα στην διεθνή αγορά. Οι επιχειρήσεις αυτές από τη μία παρέχουν τυποποιημένα προϊόντα με χαμηλή ποιότητα υπηρεσίας σε μεγάλη μάζα πελατών μέσω ενός ανοικτού ανταγωνιστικού περιβάλλοντος, ενώ από την άλλη παρέχουν σχεδιασμό και υλοποίηση εξειδικευμένων λύσεων σε συγκεκριμένους πελάτες μιας κλειστής αγοράς. Τα τυποποιημένα προϊόντα software διανέμονται μέσω licenses για τη χρήση από τον χρήστη, ενώ τα εξειδικευμένα μεταφέρονται από τον πωλητή/ παραγωγό στον αγοραστή. Πολύ λίγες εταιρίες καταφέρνουν να καλύψουν και τις δύο ομάδες πελατών. Ακόμα, είναι αρκετά συχνό φαινόμενο οι εταιρίες υψηλής τεχνολογίας να μετατοπίζουν την εστίαση τους από υπηρεσίες σε προϊόντα ή το αντίστροφο, κατά τα πρώτα χρόνια της ανάπτυξης τους. Πάντως, όπως σε όλες τις εταιρίες έτσι και στις συγκεκριμένες η στρατηγική που ακολουθείται έχει σοβαρό αντίκτυπο στο μάρκετινγκ (Ruokonen, 2004).

Ανάγκη για εξειδικευμένο και πρωτότυπο λογισμικό

Η διαμόρφωση ενός προϊόντος είναι ένας πολύ σημαντικός παράγοντας στις σχέσεις επιχείρησης προς επιχείρηση. Στις αγορές προϊόντων βασισμένων στην τεχνολογία η γρήγορη και συνεχής βελτίωση της τεχνολογίας έχει αυξήσει την ζήτηση για εξειδικευμένα προϊόντα. Στη δημιουργία της αναγκαιότητας αυτής συνετέλεσε η ανάπτυξη στις τεχνολογίες πληροφοριών, τηλεπικοινωνιών και αυτοματισμού παραγωγής.

Οι αγοραστές- επιχειρήσεις βρίσκουν τα εξειδικευμένα προϊόντα συμφέροντα όχι μόνο γιατί εξυπηρετούν καλύτερα τις συγκεκριμένες ανάγκες τους, αλλά και γιατί τα εξειδικευμένα προϊόντα συχνά ενσωματώνουν τις πιο σύγχρονες τεχνολογίες, πράγμα

που μπορεί να αποτελέσει ένα σημαντικό ανταγωνιστικό πλεονέκτημα συγκριτικά με τα τυποποιημένα προϊόντα που απευθύνονται στο ευρύτερο κοινό.

Όσον αφορά τις εταιρίες που προσφέρουν εξειδικευμένα προϊόντα, αυτό μπορεί να αποδειχθεί σημαντικό πλεονέκτημα, εφόσον η ικανότητα να αναπτύξει η εταιρία τέτοια προϊόντα μπορεί να επισημανθεί ως μία διακεκριμένη δεξιότητα και να της προσφέρει επιπρόσθετη αξία. (Stump, Athaide and Joshi, 2003).

Επιπλέον, στις εταιρίες-παραγωγούς κατ' αυτό τον τρόπο δημιουργείται γνώση και μειώνεται η αβεβαιότητα και έτσι αυξάνεται η πιθανότητα επιτυχίας του προϊόντος λογισμικού. Η αναζήτηση γνώσης μπορεί να σχετιστεί με την ταξινόμηση των απαιτήσεων, της βιωσιμότητας, της αποδοχής των χρηστών, την εμπορικότητα, τη συμπεριφορά του συστήματος ή και άλλους σημαντικούς παράγοντες απόδοσης. Πιο γενικά, η πρωτοτυπία ίσως θεωρηθεί μείωση του ρίσκου που υπάρχει από την ελλιπή γνώση των απαιτήσεων και των τρόπων επίτευξής των. Ένα εύκολο μοντέλο για τη δημιουργία καινοτομίας είναι το εξής (Tate, 1990):

- Συλλογή απαιτήσεων
- Γρήγορος σχεδιασμός
- Δημιουργία καινοτομίας
- Αξιολόγηση και τελειοποίηση σύμφωνα με τις απαιτήσεις των καταναλωτών.

Έτσι, συμπεραίνουμε πως η στρατηγική διαφοροποίησης στοχεύει στην απόκτηση ανταγωνιστικού πλεονεκτήματος καινοτομίας σε σχέση με τις άλλες επιχειρήσεις οι οποίες δραστηριοποιούνται στον ίδιο κλάδο. Το πλεονέκτημα δημιουργείται από την παραγωγή και διάθεση των προϊόντων λογισμικού ή υπηρεσιών, τα οποία γίνονται αντιληπτά από τους καταναλωτές ως μοναδικά και ποιοτικά. Οι επιχειρήσεις, οι οποίες εφαρμόζουν αυτήν τη στρατηγική, λόγω της υψηλής διαφοροποίησης έχουν τη δυνατότητα να τα τιμολογούν σε υψηλότερη τιμή από τη μέση τιμή της αγοράς και σαφώς η τιμή αυτή είναι υψηλότερη από την τιμή των προϊόντων της επιχείρησης που είναι ηγέτης κόστους (Σπαής, 2007, σελ 56).

Στην προσπάθεια να γίνει αντιληπτό στον παραγωγό τι ζητά ο καταναλωτής, υπάρχουν μερικές ομαδοποιήσεις χαρακτηριστικών που προσδιορίζουν κάθε απαίτηση. Αυτά τα χαρακτηριστικά είναι τα εξής (Mohamed and Wahba, 2008):

- Κρισιμότητα. Αναφέρεται στον βαθμό που είναι κρίσιμο το συγκεκριμένο χαρακτηριστικό για τον ενδιαφερόμενο φορέα. Οι κρίσιμες δραστηριότητες απαιτούν ειδική προσοχή μέσω του κύκλου ζωής του προϊόντος.
- Πίεση χρόνου. Αναφέρεται στο βαθμό, που είναι επείγουσα η αναγκαία απαίτηση. Στις επείγουσες απαιτήσεις δίνεται μεγαλύτερη προτεραιότητα μέσω του κύκλου ζωής του προϊόντος.
- Σταθερότητα. Αναφέρεται στον βαθμό που οι αλλαγές επηρεάζουν την απαίτηση.
- Κωδικοποίηση. Δεν είναι ανοικτού κώδικα.
- Εφικτότητα. Οι απαιτήσεις είναι υλοποιήσιμες.
- Συνέπεια. Η μη σύγκρουση διαφορετικών απαιτήσεων.
- Ορθότητα. Το προϊόν ικανοποιεί όλες τις απαιτήσεις της επιχειρηματικής ανάγκης των ενδιαφερόμενων φορέων.
- Πληρότητα. Οι απαιτήσεις είναι εκπληρωμένες και τεκμηριωμένες, χωρίς να υπολείπεται τίποτα.
- Επαληθευσιμότητα. Η υλοποίηση της απαίτησης μπορεί να γίνει μέσω ελέγχου ή ανάλυσης σχετικά με το αν υλοποιούνται οι απαιτήσεις του ενδιαφερόμενου φορέα.
- Σαφήνεια και κατανοητότητα. Τα χαρακτηριστικά έχουν την ίδια ερμηνεία από όλους τους ενδιαφερόμενους φορείς. Ο ορισμός της κάθε απαίτησης γίνεται σύντομα και περιεκτικά, χωρίς περιττές εκφράσεις ή ρήματα/προθέσεις που μπορούν να ερμηνευτούν διαφορετικά.
- Συνεκτικότητα.
- Παρατηρησιμότητα. Κάθε απαίτηση γίνεται ένα χαρακτηριστικό του προϊόντος, αντιληπτό από το χρήστη.
- Καταλληλότητα. Η ιδιότητά υλοποίησης των απαιτήσεων.

Δυσκολίες στην παραγωγή λογισμικού

Σύμφωνα με τους Yates και Hennel το 1982 οι δυσκολίες στην παραγωγή λογισμικού χωρίζονται σε δυσκολίες του μάνατζερ και δυσκολίες του προγραμματιστή.

Από την οπτική του μάνατζερ:

- Ανθρώπινοι πόροι, όπως προγραμματιστές χαμηλής ποιότητας ή έλλειψη απαραίτητης εμπειρίας.
- Οικονομικά, όπως το κόστος του κάθε πράγματος.
- Απαιτήσεις των χρηστών, δηλαδή αλλαγές απαιτήσεων ή κατανόηση των αναγκών τους.
- Αξιοπιστία λογισμικού, όπως αξιοπιστία της εφαρμογής ή έλεγχος όλων των παραμέτρων του λογισμικού.
- Πρόβλεψη, όπως πρόβλεψη πόρων ή χρόνος ολοκλήρωσης του έργου.
- Προτεραιότητα χρονοπρογραμματισμού του έργου, όπως κατανομή πόρων.
- Έλεγχος και παρακολούθηση λειτουργίας του έργου, όπως ολοκλήρωση του έργου στον ζητούμενο χρόνο, καθώς και έλεγχός του.
- Έλεγχος του λογισμικού, δηλαδή έλεγχος της ορθής λειτουργίας ή συνέπεια ως προς τα ορόσημα.
- Τεκμηρίωση, όπως ανεπαρκής τεκμηρίωση του προγράμματος.
- Ταχύτητα ή χρόνος, όπως απαιτούμενος χρόνος ολοκλήρωσης του έργου.
- Άλλα , όπως εκπαίδευση των ειδικών.

Από την οπτική του προγραμματιστή:

- Δοκιμή λογισμικού, όπως διαθεσιμότητα, προετοιμασία των δοκιμών δεδομένων ή ακόμα έλλειψη επαρκούς χρόνου για σωστό έλεγχο.
- Προδιαγραφές που θέτουν οι χρήστες, όπως ανεπαρκείς πληροφορίες ή αλλαγές επιθυμιών.

- Επικοινωνία μεταξύ αναλυτών και προγραμματιστών ή παρεμβολές της διοίκησης.
- Τεκμηρίωση, δηλαδή ποια θεωρείται καλή τεκμηρίωση αλλά και η ανάγκη να γραφούν πολλά εγχειρίδια.
- Διατήρηση του λογισμικού, όπως η διατήρηση του προγράμματος ή η συμβατότητά του με παλαιότερα συστήματα.
- Πίεση, δηλαδή η συνεχής εναλλαγή μεταξύ έργων αλλά και το κίνητρο για δημιουργία.
- Διαθέσιμο λογισμικό, όπως η χρησιμοποίηση/ και εκμάθηση νέου λογισμικού/ γλωσσών προγραμματισμού.
- Αλλά και διάφορα άλλα προβλήματα όπως η υποτίμηση του έργου ή η ανεπαρκής εμπειρία.

Στρατηγικές που ακολουθούν οι επιχειρήσεις ανάλογα με την θέση τους στον κλάδο

Υπάρχουν τριών ειδών απόψεις που χρησιμοποιούν οι επιχειρήσεις για να αποφασίσουν τη στρατηγική που θα ασκήσουν (Σπαής, 2007, σελ 95):

- Ορθολογική άποψη, στην οποία η στρατηγική προσεγγίζεται ως αποτέλεσμα μιας σειριακής, σχεδιασμένης αναζήτησης βέλτιστων λύσεων σε καθορισμένα προβλήματα με στρατηγική δράση, ως συνέπεια της εφαρμογής αποφάσεων γι' αυτά τα προβλήματα.
- Αυξητική ή εναρμονισμένη άποψη, στην οποία η στρατηγική προσεγγίζεται ως αποτέλεσμα μιας διοικητικής ενέργειας της εταιρικής ζωής. Η στρατηγική προσεγγίζεται ως ζήτημα εξέλιξης και αναγκών που πρέπει να κατανοηθούν, στο πλαίσιο των διευθυντικών δραστηριοτήτων που αντιμετωπίζουν ένα αβέβαιο και πολύπλοκο εξωτερικό περιβάλλον καθώς και ένα πλαίσιο εργασίας που χαρακτηρίζεται από κοινωνικούς και πολιτικούς παράγοντες και δράση.
- Ερμηνευτική άποψη, στην οποία η στρατηγική προσεγγίζεται ως το αποτέλεσμα ενός ατόμου ή μιας ομάδας, όπου λαμβάνονται υπόψη η

επιχείρηση και το περιβάλλον, στο οποίο πρόκειται να λειτουργήσει.

Ο παρακάτω πίνακας δίνει μια γενική εικόνα των χαρακτηριστικών που οι επιχειρήσεις παραγωγής λογισμικού μπορούν να ακολουθήσουν ανάλογα με το μέγεθος και την θέση τους στον κλάδο (Igel and Islam, 2001):

Πίνακας 1.

Στρατηγικές επιχειρήσεων (Igel and Islam, 2001).

Τεχνολογικές στρατηγικές/ θέματα	Στρατηγική ηγέτη (leader)	Στρατηγική ακόλουθου (follower)	Στρατηγική εκμεταλλευτή (exploiter)	Στρατηγική επέκτασης (extender)
Επιχειρησιακ ές στρατηγικές	Εκμετάλλευση εικόνας	Βελτίωση χαρακτηριστικ ών	Μεγιστοποίησ η ποιότητας	Ελαχιστοποίησ η τιμής
Μοναδικότητ α	Μοναδικότητα στην παραγωγή	Μοναδικότητα στον σχεδιασμό	Μοναδικότητα στην τροποποίηση	Μοναδικότητα στην πώληση
Δίνει έμφαση	Τεχνολογία know-why	Τεχνολογία physical	Διαδικασίες	Τεχνολογία Know-how
Διαδικασίες μάθησης	Παρατηρώντας και σκέπτοντας	Αλλάζοντας	Αλλάζοντας	Κάνοντας
Σκοπός	Μόχλευση (leverage)	Τέντωμα (Stretch)	Τέντωμα (Stretch)	Να εδραιωθεί (fit)

Ωστόσο, ανεξαρτήτως της θέσης που έχει η επιχείρηση στην αγορά, πρέπει να προσέξει κάποια άλλα πράγματα. Η θεωρία του Porter εφαρμόζεται και στα προϊόντα λογισμικού (Chastek, Donohoe and McGregor, 2007):

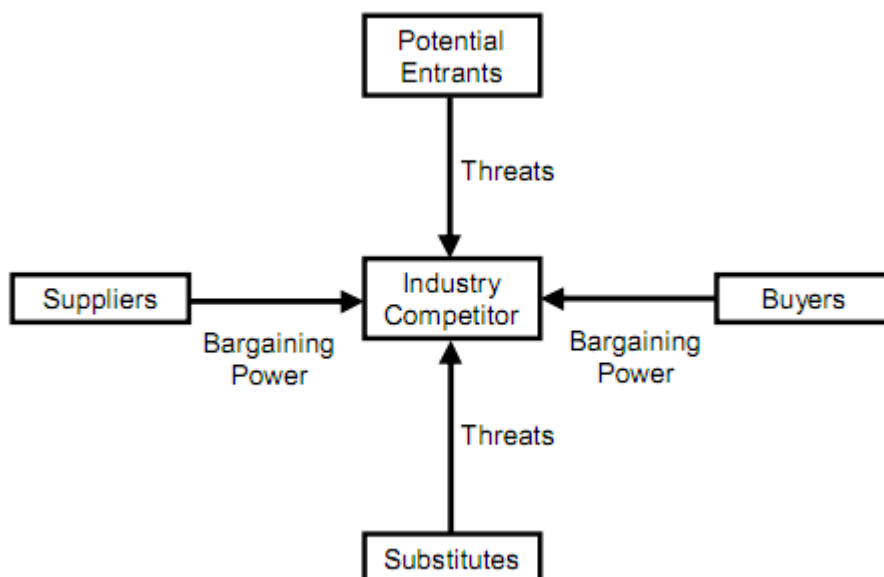
- Η πιθανότητα εισόδου ανταγωνιστών στην αγορά και η εμπόδιση του κατά το δυνατόν.
- Τα ανταγωνιστικά προϊόντα, δηλαδή να μπορέσει να μειώσει την επιθυμία για τα ανταγωνιστικά προϊόντα, μέσω των επιπρόσθετων χαρακτηριστικών που αυτή προσφέρει.
- Οι καταναλωτές, δηλαδή να μπορεί να ικανοποιεί τις ανάγκες τους.
- Οι προμηθευτές, δηλαδή η χρησιμοποίηση προγραμμάτων ανοικτού

κώδικα ώστε να καθίσταται δύσκολος ο εγκλωβισμός της σε κάποιον προμηθευτή και να έχει παραπάνω από μία αλυσίδα τροφοδοσίας.

- Οι ανταγωνιστικές επιχειρήσεις, δηλαδή η επιχείρηση να μπορεί να επωφεληθεί από τις αδυναμίες των αντιπάλων της, και να εστιάζει σε αγορά με μικρούς- αδύναμους ανταγωνιστές.

Εικόνα 2.

Γραφική απεικόνιση της θεωρίας του Porter στην βιομηχανία λογισμικού (Chastek, Donohoe and McGregor, 2007).



Στρατηγικές αποφάσεις που ακολουθούν μικρές εταιρίες παραγωγής λογισμικού

Η επιτυχία στις εταιρίες παραγωγής λογισμικού δεν απαιτεί μόνο την ανάπτυξη προϊόντων με υψηλή τεχνική ποιότητα, αλλά και άλλα στοιχεία όπως την παραγωγή των προϊόντων στην κατάλληλη χρονική στιγμή, με τα σωστά χαρακτηριστικά. Πανάκεια για την επιτυχία δεν υπάρχει καθώς (Barczak, 1995, Vahaniitty, 2003):

- Στην πράξη η θεωρητική στρατηγική ενός οικονομολόγου δεν είναι απαραίτητα καλύτερη από κάποιου άλλου.

- Η απόδοση μέσα στην εταιρία εξαρτάται από την συνεργασία μεταξύ των διατμηματικών ομάδων.
- Οι σπόνσορες του έργου συνεισφέρουν σημαντικά στην απόδοσή του, μέσα από τις ενέργειες τους.
- Η θεωρία δεν μπορεί να εφαρμοστεί ευθέως σε μικρές εταιρίες, εφόσον τις περισσότερες φορές σχεδιάστηκε για μεγάλες. Παρ' όλα αυτά, πρέπει να υπάρχει προγραμματισμός σε μια μικρή εταιρία και οι στόχοι τις να είναι σαφείς.
- Υπάρχουν τεχνικές, εργαλεία και μέθοδοι για την ευθυγράμμιση της ανάπτυξης νέων προϊόντων με την στρατηγική. Αυτά όμως έγιναν για μεγάλες εταιρίες, με δυνατότητα μεγάλης παραγωγής και μεγάλο εργατικό δυναμικό. Δεν μπορούν δηλαδή να εφαρμοστούν αυτούσιες στις μικρές επιχειρήσεις.
- Η πλειοψηφία της βιβλιογραφίας βρίσκεται υπό το πρίσμα μεγάλων οργανισμών και επιχειρήσεων που έκαναν ειδικά έργα για συγκεκριμένους πελάτες.

Στη συνέχεια της εργασίας θα αναφερθούν οι σημαντικότερες από τις αποφάσεις που πρέπει να πάρει μία μικρή επιχείρηση παραγωγής λογισμικού (Vahaniitty, 2003):

- Διαχείριση χαρτοφυλακίου (Portfolio management). Περιλαμβάνει τη διαμόρφωση και εφαρμογή στρατηγικών για προϊόντα σε ολόκληρη την γκάμα προϊόντων και υπηρεσιών που προσφέρονται από την επιχείρηση, με βάση τις απαιτήσεις και τους περιορισμούς και τις απαιτήσεις που τέθηκαν από τις υπόλοιπες συνιστώσες του επιχειρησιακού μοντέλου.
- Οργάνωση (Organization). Εδώ είναι απαραίτητη η ύπαρξη οργανογράμματος, ρόλων και ευθυνών, στελέχωση ομάδας, φυσική διάταξη της ομάδας, ικανότητες, χρήση εξωτερικών πηγών, υποδομές ανάπτυξης.
- Μοντέλο ανάπτυξης (Development model).

- ✓ Ρυθμός ανάπτυξης,
- ✓ Έλεγχος ανάπτυξης (Παρακολούθηση και έλεγχος προόδου, Μηχανισμοί επικοινωνίας, Σχετική προτεραιότητα και αλληλεπίδραση με άλλα μοντέλα ανάπτυξης).
- Διαχείριση έργου (Product management)
- ✓ Επιλογή τεχνολογίας
- ✓ Τεχνολογικές απαιτήσεις
- ✓ Διαχείριση αλλαγών
- ✓ Παράδοση και διαμόρφωση
- Στρατηγική ποιότητας (Quality strategy)
- ✓ Ορισμός αποδεκτής ποιότητας
- ✓ Διαχείριση κινδύνου

Βέλτιστος χρόνος για την έκδοση νέου λογισμικού

Σύμφωνα με την μελέτη των Levin και Yadid το 1990 οι τέσσερις βασικοί λόγοι κόστους για την έκδοση ενός νέου λογισμικού είναι οι εξής:

- Το καθορισμένο κόστος λόγω τεκμηρίωσης, διανομής, εγκατάστασης και εκπαίδευσης.
- Το κόστος διόρθωσης σφαλμάτων.
- Το κόστος βελτιώσεων και ενισχύσεων.
- Το κόστος λόγω απόσυρσης του λογισμικού.

Με βάση τους τέσσερις αυτούς παράγοντες, το κόστος λειτουργίας του προϊόντος συγκρίνεται με το συνολικό κόστος νέας κυκλοφορίας, προκειμένου να καθοριστεί ο βέλτιστος χρόνος για τη νέα κυκλοφορία.. Τα αποτελέσματα της μελέτης μπορούν να χωριστούν σε τρεις ομάδες: α) Το συνολικό κόστος λειτουργίας ελαχιστοποιείται σε ένα σημείο στο μέλλον, το χαμηλότερο αυτό σημείο της λειτουργίας είναι το ιδανικότερο σημείο για την επόμενη έκδοση, β) Το συνολικό κόστος λειτουργίας

μειώνεται με τον χρόνο, που σημαίνει πως η νέα έκδοση πρέπει να αναβάλλεται επ' αόριστον, γ) Το συνολικό κόστος λειτουργίας αυξάνεται με τον χρόνο, άρα πρέπει συνεχώς να υπάρχουν νέες εκδόσεις του λογισμικού.

Πρόβλημα αυτής της μεθόδου αποτελεί η μυωπική της φύση, αδυνατεί δηλαδή να αξιολογήσει εξωτερικούς παράγοντες όπως τον ανταγωνισμό ή εσωτερικά κόστη όπως το κόστος προγραμματισμού. Μία εταιρία παραγωγής λογισμικού το πιο πιθανό είναι να θέλει να ελαχιστοποιεί το συνολικό κόστος του κύκλου ζωής του προϊόντος (Levin and Yadid, 1990).

Οι δέκα πιο συχνά εμφανιζόμενοι κίνδυνοι στην κατασκευή προϊόντων λογισμικού είναι οι εξής, σύμφωνα με τη σειρά πιο πιθανούς εμφάνισης (Han; Huang, 2006):

1. Συνεχής αλλαγή των απαιτήσεων του συστήματος.
2. Όχι επαρκώς προσδιορισμένες απαιτήσεις.
3. Ασαφής απαιτήσεις του συστήματος.
4. Έλλειψη αποτελεσματικής μεθοδολογίας διαχείρισης έργου.
5. Εσφαλμένες απαιτήσεις συστήματος.
6. Φτωχό πλάνο του έργου.
7. Ανεπαρκής εκτίμηση των απαιτούμενων πόρων.
8. Χρησιμοποίηση νέας/ άγνωστης τεχνολογίας.
9. Όχι αρκετά καλή παρακολούθηση της προόδου του έργου.
10. Εταιρικής πολιτική με αρνητικές συνέπειες για το έργο.

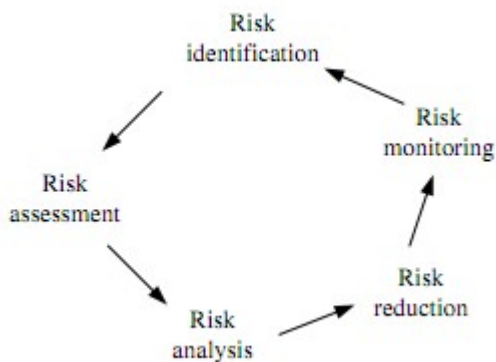
Προκειμένου να μελετηθούν τα πιθανά ρίσκα για την ανάπτυξη ή βελτίωση ενός προϊόντος οι διεργασίες που χρειάζονται είναι οι εξής (Xie, Zhang and Lai, 2005):

- Εντοπισμός κινδύνου. Μία ομάδα που εντοπίζει τους κινδύνους σε κάθε φάση του κύκλου ζωής.
- Αξιολόγηση κινδύνου, σύμφωνα με τον τύπο του Boehm $RE = P(UO) * L(UO)$, όπου $P(UO)$ η πιθανότητα μιας δυσάρεστης έκβασης, $L(UO)$ το κόστος της έκβασης αυτής και RE ο κίνδυνος.

- Ανάλυση κινδύνου. Το σημαντικό είναι η ενσωμάτωση των κινδύνων του έργου στους κινδύνους της προσφοράς.
- Μείωση κινδύνου. Εξαρτάται από την ανάλυση κινδύνου, την αντίδραση σε αυτόν και την αποφυγή του.
- Παρακολούθηση κινδύνου. Στην φάση αυτή πρέπει να προσεχθεί ιδιαίτερα η μεταμόρφωση ενός κινδύνου και οι διάφορες συσχετίσεις μεταξύ διαφορετικών κινδύνων.

Εικόνα3.

Οι πέντε διεργασίες για τον εντοπισμό ρίσκου (Xie, Zhang and Lai, 2005).



Αξιολόγηση λογισμικού

Η διαδικασία αξιολόγησης λογισμικού βασίζεται στην διοίκηση ολικής ποιότητας. Ο βασικότερος λόγος της αξιολόγησης είναι η ανάπτυξη ή απόκτηση καλού προϊόντος λογισμικού. Έτσι, μπορεί να υπάρχει αξιολόγηση από (Azuma, 1996):

1. Κατασκευαστές. Είτε για να κριθεί ο σωστός χρόνος για την απελευθέρωση του προϊόντος, είτε για να συγκριθεί με ανταγωνιστικά προϊόντα.
2. Αγοραστές. Σκοπός της αξιολόγησης από τους αγοραστές είναι η επιλογή του βέλτιστου δυνατού προϊόντος. Επίσης, μέσω αυτής της αξιολόγησης οι αγοραστές μπορούν να κρίνουν εάν το συγκεκριμένο προϊόν ικανοποιεί τις απαιτήσεις που έχει ο καταναλωτής. Έτσι, η αξιολόγηση συνήθως γίνεται συγκρίνοντας τις προδιαγραφές. Για κρίσιμα προϊόντα λογισμικού, η διασφάλιση της ποιότητας είναι απαραίτητο να ελέγχεται και από τον

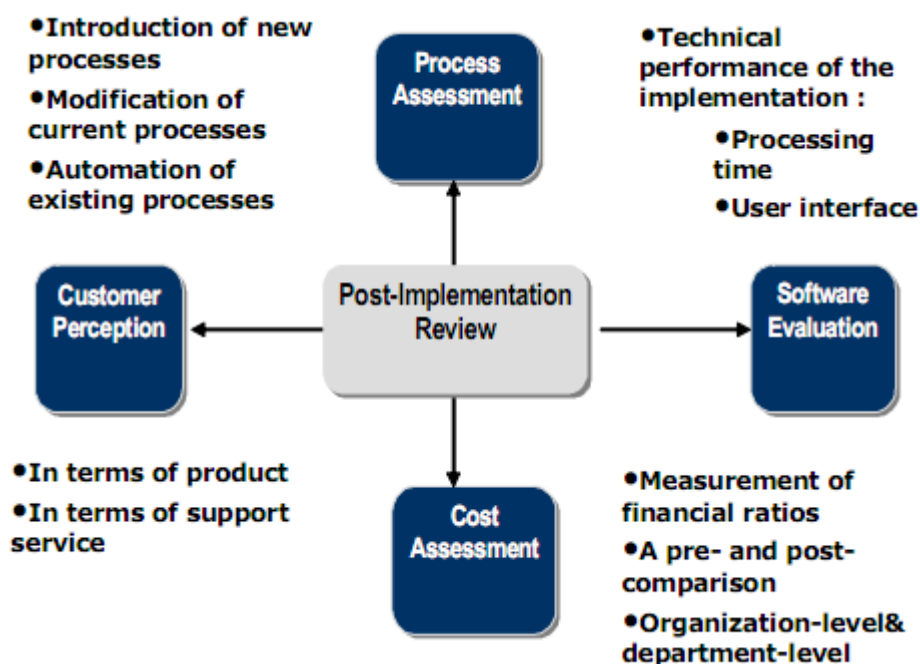
προμηθευτή και από τον καταναλωτή, ακόμα και εάν κάποιες απαιτήσεις ποιότητας δεν είναι δηλωμένες.

3. Ανεξάρτητους αξιολογητές. Σημαίνει αξιολόγηση του προϊόντος από ανεξάρτητους αξιολογητές είτε από αίτημα του κατασκευαστή, είτε από αίτημα του αγοραστή, είτε για κάποιον τρίτο όπως η σύγκριση προϊόντων λογισμικού για ένα περιοδικό.

Στο παρακάτω σχήμα φαίνονται τα κριτήρια, με τα οποία γίνεται μία έρευνα για την αξιολόγηση ενός προϊόντος λογισμικού. Η αξιολόγηση γίνεται μέσω τεσσάρων οπτικών, που είναι η διαδικασία αξιολόγησης, το κόστος αξιολόγησης, η τεχνική αξιολόγηση και η αντίληψη του πελάτη. Ανάλογα το σύστημα και τον οργανισμό, η αξιολόγηση πρέπει να γίνεται μετά από έξι μήνες με ένα χρόνο από την έκδοσή του. (Kakkar, 2006).

Εικόνα 4.

Κριτήρια αξιολόγησης προϊόντων λογισμικού(Kakkar, 2006).



Συμπεράσματα

Στον κλάδο των επιχειρήσεων δεν υπάρχουν ούτε “από μηχανής θεοί” ούτε βέλτιστες στρατηγικές, που αν ακολουθηθούν, αυτόματα θα λύσουν όλα τα προβλήματα. Οι στρατηγικές που πρέπει να ακολουθήσει μία επιχείρηση πρέπει να ληφθούν κατόπιν σοβαρής σκέψης και μέσω διοικητικών αποφάσεων τόσο για το βραχυπρόθεσμο, όσο και το μεσοπρόθεσμο και μακροπρόθεσμο διάστημα. Δηλαδή, σε επίπεδο μικροοικονομίας, αλλά και μακροοικονομίας.

Στην Ελλάδα, στην πλειοψηφία τους οι νέες επιχειρήσεις, ιδιαίτερα εξαιτίας των οικονομικών συνθηκών που επικρατούν σήμερα, είναι μικρές επιχειρήσεις. Άρα, πρέπει τουλάχιστον στην αρχή να αντιμετωπισθούν σαν τέτοιες. Έτσι, μία νέα, μικρή εταιρία πρέπει να διαχειριστεί το χαρτοφυλάκιο της, να δημιουργήσει το οργανόγραμμα και το μοντέλο ανάπτυξής της και να παράγει κάθε νέο προϊόν κατόπιν της διαδικασίας διαχείρισης έργου, με υψηλά στάνταρ ποιότητας.

Προτάσεις για μελλοντική έρευνα

Η εργασία αυτή θα μπορούσε να χρησιμοποιηθεί από μία επιχείρηση που επιδιώκει τη δραστηριοποίησή της στον τομέα της ανάπτυξης λογισμικού. Κι αυτό γιατί αναφέρει όλες τις ιδιαιτερότητες των προϊόντων λογισμικού, καθώς και τις στρατηγικές που χρησιμοποιούνται, σύμφωνα με διάφορους οικονομολόγους, από τις αντίστοιχες επιχειρήσεις, ανάλογα με τις βλέψεις τους και τη θέση τους στον κλάδο. Βέβαια,, κάθε φορά που μία επιχείρηση θέλει να ξεκινήσει τις δραστηριότητές της, πρέπει να κάνει μία κλαδική μελέτη για να διαπιστώνονται οι τάσεις της αγοράς, η αγοραστική ισχύς των καταναλωτών και ο ανταγωνισμός στη συγκεκριμένη χρονική περίοδο.

Βιβλιογραφία

- 1) Zangwill, W., (1993). Lighting Strategies for Innovation (10^η έκδοση). New York: Lexington Books.
- 2) Σπαής, Γ., (2007). Εισαγωγή στη διαχείριση τεχνολογικών καινοτομιών (1^η έκδοση). Αθήνα: Εκδόσεις Κριτική Α.Ε.
- 3) Aramand, M.,(2008). Software products and services are high tech? New product development strategy for software products and services. Technovation. 28 (3), 154-160.
- 4) Tyrvaenen, P., Warsta, J., Seppanen, V., (2008). Evolution of secondary software business: Understanding industry dynamics. IFIP International Federation for Information Processing. 287, 381-401.
- 5) Ruokonen. M., (2004). Market orientation and product strategies in small internationalizing software companies. Internet Small Journal. 18 (2), 143-156.
- 6) Frambach, R., Prabhu, J., Verhallen, T., (2003). The influence of business strategy on new product activity: The role of market orientation. International Journal Of Research in Marketing. 20 (4), 377-397.
- 7) Igel, B., Islam N., (2001). Strategies for service and market development of entrepreneurial software designing firms. Technovation. 21 (3), 157-166.
- 8) Stump, R., Athaide, G., Joshi, A., (2003). Managing seller-buyer new product development relationships for customized products: a contingency model based on transaction cost analysis and empirical test. Journal of Product Innovation Management. 19 (6), 439-454.
- 9) Barczak,G., (1995). New Product Strategy; Structure, Process, and Performance in the Telecommunications industry. Journal of Product Innovation Management. 12 (3), 224-234.
- 10) Yates, D., Hennel, M., (1982). An examination of standards and practices for software production; Computers and Standards. 1 (1 and 2), 3-41.
- 11) Levin, K., Yadid, O., (1990). Optimal realease time of improved versions of software packages. Information and software technology. 32(1), 65-70.
- 12) Mohamed, S., Wahba, A.,(2008). Value estimation for software product management. IEEM. 2196-2200.
- 13) Veldhuizen, E., Hulting, E., Griffin, A., (2006). Modeling market information processing in new product development: An empirical analysis. Journal of

engineering and technology management. 23(4), 353-373.

- 14) Tate, G., (1990). Prototyping: helping to build the right software. Information and software technology. 32 (4), 237-44.
- 15) Xie, G., Zhang, J., Lai, K., (2005). Risk avoidance in bidding for software projects based on life cycle management theory. International journal of project management. 24 (6), 516-521.
- 16) Han, W., Huang, S., (2005). An empirical analysis of risk components and performance on software projects. Journal of systems and software. 80 (1), 42-50.
- 17) Azuma, M., (1996). Software products evaluation system: quality models, metrics and process- International Standards and Japanese Practise. Information and Software Technology. 38 (3), 145-154.
- 18) Vahaniitty, J., (2003, September 1-6). Key decisions in strategic new product development for small software product businesses. Euromicro Conference, Espoo, Finland, p.373-383.
- 19) Chastek, G., Donohoe, P., McGregor, J., (2007, September 10-14). A production system for software product lines. 11th International Software Product Line Conference, Kyoto, Japan, p.117-128.
- 20) Kakkar, S., (2006, September). Implementation aspects of software development projects. India Conference. New Delhi, India, p.1-6.
- 21) Retrieve 10 May 2011, from http://en.wikipedia.org/wiki/Software_development.