
Software Evolution Monitor

Software Evolution Monitor (Requirements Definition)

Έκδοση <1.0>

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

Ιστορικό Προηγούμενων Εκδόσεων

Ημερομηνία	Έκδοση	Περιγραφή	Συγγραφέας
<11/09/2014>	<1.0>	1 ^η έκδοση της περιγραφής των απαιτήσεων.	A. Ζάρρας

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

Περιεχόμενα

1.	Εισαγωγή	4
1.1	Σκοπός	4
1.2	Εύρος	6
1.3	Αναφορές	6
1.4	Σύνοψη	6
2.	Μοντέλο δεδομένων	6
3.	Λειτουργικές Απαιτήσεις	8
4.	Μη Λειτουργικές Απαιτήσεις	14
5.	Απαιτήσεις Τεκμηρίωσης	15
6.	Απαιτήσεις Σχετικές με τη Διεργασία Ανάπτυξης	15
7.	Εργαλεία υλοποίησης	15

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

Περιγραφή Απαιτήσεων (Software Requirements Definition)

1. Εισαγωγή

1.1 Σκοπός

Το παρόν κείμενο περιλαμβάνει τις απαιτήσεις που συλλέχθηκαν για την κατασκευή ενός εργαλείου που έχει ως στόχο την παρακολούθηση της εξέλιξης συστημάτων λογισμικού. Γενικά, οι αποφάσεις που λαμβάνουμε και ο προγραμματισμός της όλης διαδικασίας συντήρησης ενός συστήματος λογισμικού υποβοηθούνται σημαντικά από την κατανόηση του πώς εξελίσσεται το λογισμικό με την πάροδο του χρόνου. Ενδιαφερόμαστε για τις αλλαγές που γίνονται, το μέγεθος του λογισμικού, την πολυπλοκότητά του, το ρυθμό αύξησης του μεγέθους, κ.α. Υπό ιδανικές συνθήκες, η εξέλιξη του λογισμικού δεν είναι ανεξέλεγκτη, γίνεται με ένα οργανωμένο τρόπο ο οποίος καθορίζεται από 2 αντικρουόμενες τάσεις που εξασφαλίζουν την ισορροπημένη εξέλιξη του λογισμικού. Η πρώτη (θετική) τάση αφορά σε δραστηριότητες που αυξάνουν τις λειτουργίες που προσφέρει το λογισμικό με σκοπό την ικανοποίηση των συνεχώς αυξανόμενων απαιτήσεων που έχουν οι χρήστες του λογισμικού. Η δεύτερη (αρνητική) τάση αφορά σε δραστηριότητες συντήρησης των λειτουργιών που προσφέρει ήδη το λογισμικό, οι οποίες κατ' επέκταση περιορίζουν την ανεξέλεγκτη αύξηση των λειτουργιών του λογισμικού, που προστάζει η θετική τάση εξέλιξης του λογισμικού. Η ύπαρξη των 2 αυτών τάσεων που εξασφαλίζουν την ισορροπημένη εξέλιξη του λογισμικού παρατηρήθηκε για πρώτη φορά στις εμπειρικές μελέτες που έκανε ο Meir Lehman και οι συνεργάτες τη δεκαετία του 70 σε πραγματικά συστήματα λογισμικού. Οι μελέτες αυτές οδήγησαν στη διατύπωση ενός συνόλου κανόνων που περιγράφουν την εξέλιξη του λογισμικού οι οποίοι είναι γνωστοί ως Νόμοι του Lehman. Πιο συγκεκριμένα, οι Νόμοι του Lehman στην τρέχουσα μορφή τους και μετά από 40 χρόνια περαιτέρω έρευνας είναι:

- I. **Συνεχής Αλλαγή:** Ένα σύστημα λογισμικού πρέπει να προσαρμόζεται συνεχώς στις ανάγκες των χρηστών, διαφορετικά η χρήση του γίνεται λιγότερο ικανοποιητική με την πάροδο του χρόνου.
- II. **Αυξανόμενη Πολυπλοκότητα:** Όπως ένα σύστημα λογισμικού αλλάζει με την πάροδο του χρόνου, η πολυπλοκότητά του αυξάνει, εκτός και αν γίνονται εργασίες συντήρησης για τη μείωση της.

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

- III. **Αυτορυθμιζόμενη Εξέλιξη:** Η εξέλιξη ενός συστήματος λογισμικού ρυθμίζεται από διαδικασίες ανατροφοδότησης. Υπάρχουν 2 είδη ανατροφοδότησης θετική και αρνητική. Η θετική ανατροφοδότηση περιλαμβάνει νέες απαιτήσεις από διάφορες πηγές (π.χ., χρήστες, πελάτες) που οδηγούν στην αύξηση των λειτουργιών του συστήματος. Η αρνητική ανατροφοδότηση περιλαμβάνει αναφορές σε προβλήματα που προκύπτουν (π.χ., σφάλματα, κακή σχεδίαση, υλοποίηση) και οδηγούν σε δραστηριότητες συντήρησης του λογισμικού που περιορίζουν την ανεξέλεγκτη ανάπτυξή του. Τα δύο είδη ανατροφοδότησης είναι αντικρουόμενα (με την έννοια ότι οι πόροι που είναι διαθέσιμοι για την εξέλιξη του συστήματος μοιράζονται σε δραστηριότητες που αφορούν στα 2 είδη ανατροφοδότησης) με συνέπεια η εξέλιξη να σταθεροποιείται, με την υποσημείωση ότι σταθερότητα δεν σημαίνει ότι δεν γίνονται αλλαγές, αλλά ότι γίνονται με ένα οργανωμένο τρόπο που ακολουθεί συγκεκριμένα επαναλαμβανόμενα μοτίβα.
- IV. **Διατήρηση της Εργασιακής Σταθερότητας:** Ο ρυθμός των εργασιών που γίνονται για την εξέλιξη ενός συστήματος λογισμικού τείνει να είναι σταθερός με την πάροδο του χρόνου.
- V. **Διατήρηση της Εξοικείωσης:** Η αύξηση των λειτουργιών ενός συστήματος λογισμικού περιορίζεται από την ανάγκη εξοικείωσης των χρηστών με τις αλλαγές που γίνονται στο σύστημα.
- VI. **Συνεχής Αύξηση:** Οι λειτουργίες ενός συστήματος λογισμικού αυξάνουν συνεχώς με στόχο την ικανοποίηση των αναγκών των χρηστών που μεταβάλλονται με την πάροδο του χρόνου.
- VII. **Πτωτική Ποιότητα:** Η ποιότητα ενός συστήματος λογισμικού θα εμφανίζει πτωτική τάση, εκτός και εάν στο σύστημα γίνεται εντατική συντήρηση και αναπροσαρμογή.
- VIII. **Σύστημα Ανατροφοδότησης:** Η εξέλιξη ενός συστήματος λογισμικού είναι ένας μηχανισμός ανατροφοδότησης που περιλαμβάνει πολλαπλά επίπεδα, βρόγχους και εμπλεκόμενους. Η κατανόηση και μοντελοποίηση του μηχανισμού αυτού επιτρέπει την πρόβλεψη της εξέλιξης του λογισμικού, που είναι σημαντική για την αποτελεσματική οργάνωση και διαχείριση των πόρων που απαιτούνται για την εξέλιξη του λογισμικού.

Περισσότερες λεπτομέρειες σχετικά με τους νόμους του Lehman καθώς και τον τρόπο αποτίμησης του κατά πόσον ισχύουν στο [3].

Με βάση τα παραπάνω στόχος του εργαλείου που θα αναπτυχθεί είναι η παρακολούθηση της εξέλιξης συστημάτων λογισμικού και η διευκόλυνση της αποτίμησης του κατά πόσο το σύνολο των Νόμων που διατύπωσε ο Lehman ισχύει για τα εν λόγω συστήματα λογισμικού.

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

Περεταίρω λεπτομέρειες σχετικά με τις επιμέρους λειτουργικές και μη-λειτουργικές απαιτήσεις που αφορούν στο εργαλείο δίνονται στη συνέχεια.

1.2 Εύρος

Το παρόν κείμενο αποτελεί την είσοδο στη φάση της ανάλυσης και προδιαγραφής των απαιτήσεων όπου και αναμένεται μια τυπική περιγραφή των απαιτήσεων που συλλέχθηκαν για το ζητούμενο εργαλείο και οι οποίες δίνονται στη συνέχεια.

1.3 Αναφορές

[1] S. Pfleeger. «Τεχνολογία Λογισμικού, Θεωρία και Πράξη», Κλειδάριθμος.

[2] I. Sommerville. «Εισαγωγή στην Τεχνολογία Λογισμικού», Κλειδάριθμος.

[3] A M. M. Lehman, J. F. Ramil, D. E. Perry. On Evidence Supporting the FEAST Hypothesis and the Laws of Software Evolution. Proceedings of the 5th International Symposium on Software Metrics, pages 84-88, 1998. <http://users.ece.utexas.edu/~perry/work/papers/DP-98-metrics.pdf>

1.4 Σύνοψη

Το υπόλοιπο του παρόντος κειμένου χωρίζεται σε ενότητες οι οποίες περιλαμβάνουν πιο συγκεκριμένα απαιτήσεις σχετικές με τα δεδομένα που χειρίζεται το εργαλείο (Ενότητα 2), τις λειτουργικές απαιτήσεις του ζητούμενου εργαλείου (Ενότητα 3), τις μη-λειτουργικές απαιτήσεις του ζητούμενου εργαλείου (Ενότητα 4) και περαιτέρω πληροφορίες που αφορούν στην τεκμηρίωσή του (Ενότητα 5), τη διεργασία ανάπτυξής του (Ενότητα 6) και τα εργαλεία ανάπτυξης (Ενότητα 7).

2. Μοντέλο δεδομένων

Στην ενότητα αυτή καθορίζεται ένα γενικό μοντέλο των δεδομένων που διαχειρίζεται το εργαλείο. Πιο συγκεκριμένα, για ένα σύστημα λογισμικού το εργαλείο θα δέχεται σαν είσοδο ένα αρχείο με πληροφορίες που επιτρέπουν τη σύνθεση του πλήρους ιστορικού εξέλιξης του λογισμικού. Η μορφή του αρχείου περιγράφεται στο σχήμα 1 με ένα παράδειγμα. Στην πρώτη γραμμή δίνεται το όνομα του λογισμικού. Ακολουθούν το πλήθος των λειτουργιών και το πλήθος των δομών δεδομένων που υπάρχουν στην υλοποίηση της 1^{ης} έκδοσης του λογισμικού. Στη συνέχεια δίνονται οι αλλαγές που έχουν γίνει σε κάθε μια από τις επόμενες εκδόσεις του λογισμικού. Πιο συγκεκριμένα, για κάθε έκδοση δίνεται ένας μοναδικός κωδικός ID, η ημερομηνία έκδοσης, το πλήθος των λειτουργιών που προστέθηκαν, αφαιρέθηκαν και μετατράπηκαν, καθώς και το πλήθος των δομών δεδομένων που προστέθηκαν, αφαιρέθηκαν και μετατράπηκαν.

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

```
Initial Number of Operations;21
Initial Number of Data Structures;105

ID;Date;Operations Added;Operations Deleted;Operations Updated;Data Structures Added;Data Structures Deleted;Data Structures Updated
1;13/10/2005;0;0;0;0;0;0
2;3/3/2006;2;0;9;9;0;23
3;27/4/2006;2;0;5;9;0;10
4;14/6/2006;2;0;8;12;0;19
5;30/6/2006;1;0;8;4;0;20
6;7/8/2006;0;0;0;0;0;0
7;20/9/2006;5;0;6;21;0;14
8;20/10/2006;1;0;1;5;0;4
9;1/12/2006;0;0;0;0;0;0
10;20/3/2007;1;0;4;4;0;11
11;30/4/2007;0;0;0;0;0;0
12;20/7/2007;5;0;35;21;0;36
13;4/9/2007;0;0;0;0;0;0
14;13/12/2007;0;0;0;0;0;0
15;3/3/2008;1;0;1;6;0;3
16;17/3/2008;0;0;0;0;0;0
17;9/5/2008;0;2;0;0;5;1
18;4/11/2008;2;1;1;9;2;3
19;1/12/2011;3;2;5;20;5;13
20;29/3/2012;3;0;7;14;0;17
```

Σχήμα 1.

Επομένως το ιστορικό εξέλιξης του λογισμικού χαρακτηρίζεται από το όνομα και από μια λίστα εκδόσεων του λογισμικού. Επίσης το ιστορικό εξέλιξης χαρακτηρίζεται και από μια εγγραφή στην οποία καταγράφεται το κατά πόσον ισχύει καθένας από του νόμους του Lehman. Μια έκδοση χαρακτηρίζεται από ένα ID και μια ημερομηνία. Επίσης μια έκδοση περιλαμβάνει:

- Μια εγγραφή στην οποία καταγράφονται οι αλλαγές (προσθήκες, διαγραφές, μετατροπές) που έγιναν στις λειτουργίες.
- Μια εγγραφή στην οποία καταγράφονται οι αλλαγές (προσθήκες, διαγραφές, μετατροπές) που έγιναν στις δομές δεδομένων.
- Μια εγγραφή στην οποία καταγράφονται μετρικές που αφορούν στο μέγεθος και την πολυπλοκότητα των λειτουργιών του λογισμικού. Οι μετρικές αυτές είναι:
 - **Πλήθος λειτουργιών.**
 - **Ρυθμός ανάπτυξης λειτουργιών**, που υπολογίζεται ως η διαφορά του πλήθους των λειτουργιών που προσφέρει η τρέχουσα έκδοση από το πλήθος των λειτουργιών που πρόσφερε η προηγούμενη έκδοση.
 - **Πολυπλοκότητα**, που υπολογίζεται ως το κλάσμα (πλήθος λειτουργιών που μετατράπηκαν και διαγράφηκαν) / (πλήθος λειτουργιών που προστέθηκαν).
 - **Ρυθμός εργασιών**, που υπολογίζεται ως το κλάσμα (συνολικό πλήθος λειτουργιών που προστέθηκαν, μετατράπηκαν και διαγράφηκαν) / (χρονικό διάστημα μεταξύ τρέχουσας και προηγούμενης έκδοσης).

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

- Μια εγγραφή στην οποία καταγράφονται μετρικές που αφορούν στο μέγεθος και την πολυπλοκότητα των δομών δεδομένων του λογισμικού. Οι μετρικές αυτές είναι:
 - ο **Πλήθος δομών δεδομένων**.
 - ο **Ρυθμός ανάπτυξης δομών δεδομένων**, που υπολογίζεται ως η διαφορά του πλήθους των δομών δεδομένων της τρέχουσας έκδοσης από το πλήθος των δομών δεδομένων της προηγούμενης έκδοσης.
 - ο **Πολυπλοκότητα**, που υπολογίζεται ως το κλάσμα (πλήθος δομών δεδομένων που μετατράπηκαν και διαγράφηκαν) / (πλήθος δομών δεδομένων που προστέθηκαν).
 - ο **Ρυθμός εργασιών**, που υπολογίζεται ως το κλάσμα (συνολικό πλήθος δομών δεδομένων που προστέθηκαν, μετατράπηκαν και διαγράφηκαν) / (χρονικό διάστημα μεταξύ τρέχουσας και προηγούμενης έκδοσης).

3. Λειτουργικές Απαιτήσεις

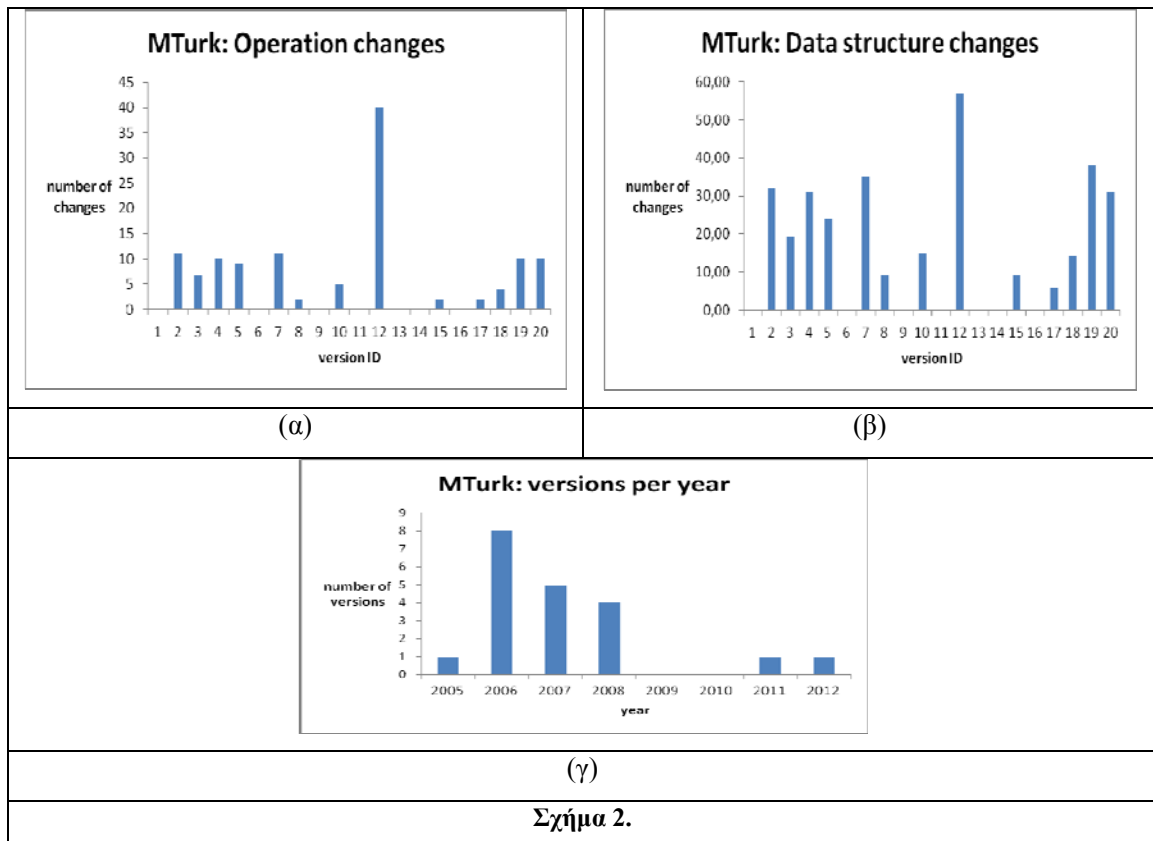
Ο χρήστης του εργαλείου χρησιμοποιεί το εργαλείο ενεργοποιώντας τα παρακάτω σενάρια χρήσης.

1. **Καταγραφή ιστορικού εξέλιξης:** Ο χρήστης καθορίζει το αρχείο που περιλαμβάνει πληροφορίες για την εξέλιξη ενός συστήματος λογισμικού. Ακολουθώντας, με βάση τα δεδομένα του αρχείου δημιουργείται το ιστορικό εξέλιξης του λογισμικού. Η διαδικασία αυτή μπορεί να επαναληφθεί πολλές φορές για διαφορετικά συστήματα λογισμικού. Το εργαλείο οφείλει να διαχειρίζεται ταυτόχρονα το ιστορικό εξέλιξης περισσότερων του ενός συστημάτων λογισμικού. Στην περίπτωση που ο χρήστης επιχειρήσει να δώσει σαν είσοδο στο εργαλείο το ίδιο αρχείο, το εργαλείο εμφανίζει μήνυμα λάθους. Στην περίπτωση που το αρχείο που δίνει ο χρήστης δεν υπάρχει ή δεν έχει την αναμενόμενη μορφή το εργαλείο εμφανίζει μήνυμα λάθους.
2. **Αποτίμηση του 1^{ου} νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολουθώντας προχωρά στην διαδικασία αποτίμησης του 1^{ου} νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα ραβδόγραμμα (bar chart) που απεικονίζει το πλήθος των αλλαγών που έχουν γίνει σε λειτουργίες σε κάθε έκδοση του λογισμικού (π.χ. Σχ. 2(α)). Επίσης το εργαλείο εμφανίζει στο χρήστη ένα ραβδόγραμμα (bar chart) που απεικονίζει το πλήθος των αλλαγών που έχουν γίνει σε δομές δεδομένων σε κάθε έκδοση του λογισμικού (π.χ. Σχ. 2(β)). Τέλος, το εργαλείο εμφανίζει στο χρήστη ένα ραβδόγραμμα (bar chart) που απεικονίζει το πλήθος των εκδόσεων του λογισμικού ανά έτος (π.χ. Σχ. 2(γ)). Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν
 - a. σε κάθε έκδοση του λογισμικού υπάρχουν αλλαγές στις λειτουργίες ή στις δομές δεδομένων του εργαλείου και
 - b. κάθε χρόνο υπάρχει τουλάχιστον μια έκδοση του λογισμικού.

Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.



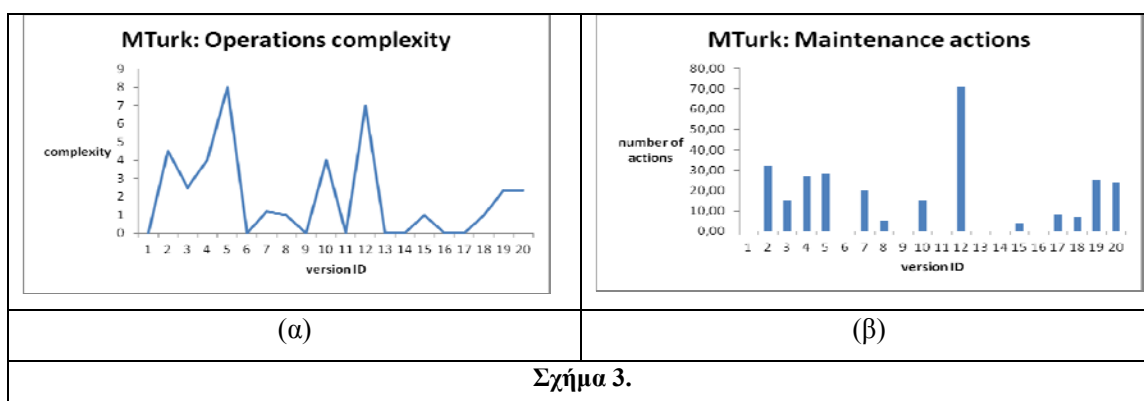
3. **Αποτίμηση του 2ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 2^{ου} νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει την πολυπλοκότητα των λειτουργιών του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 3(α)) και ένα γράφημα γραμμών (line chart) που απεικονίζει την πολυπλοκότητα των δομών δεδομένων του λογισμικού σε κάθε έκδοση. Τέλος το εργαλείο εμφανίζει στο χρήστη ένα ραβδόγραμμα που απεικονίζει το πλήθος των δραστηριοτήτων συντήρησης (διαγραφές και μετατροπές στις λειτουργίες και στις δομές δεδομένων) που έχουν γίνει σε κάθε έκδοση (π.χ. Σχ. 3(β)). Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν

- a. η πολυπλοκότητα των λειτουργιών ή των δομών δεδομένων τείνει να αυξάνεται ή

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

- b. η πολυπλοκότητα των λειτουργιών και των δομών δεδομένων δεν έχει αυξητική τάση και επιπλέον στο ραβδόγραμμα υπάρχουν εμφανείς δραστηριότητες συντήρησης.

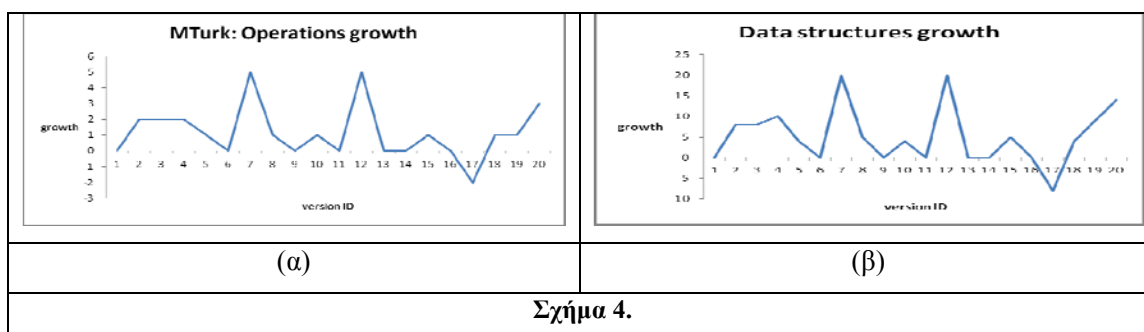
Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.



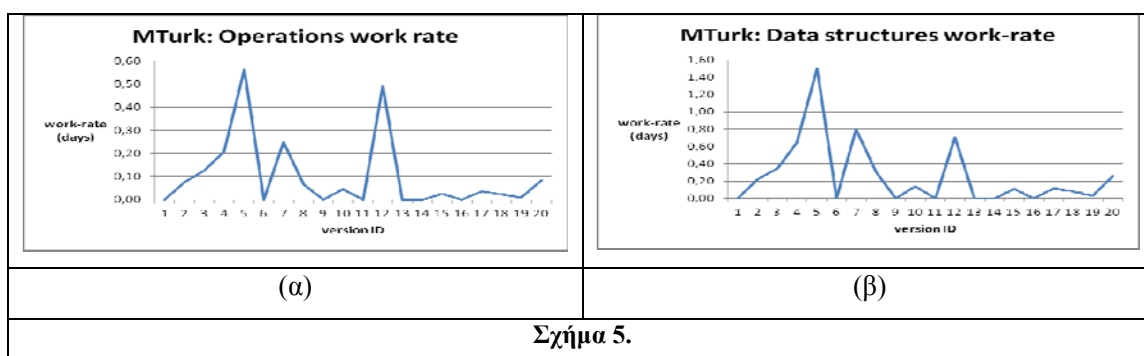
4. **Αποτίμηση του 3ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 3ου νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει το ρυθμό ανάπτυξης λειτουργιών (π.χ. Σχ. 4(α)) του λογισμικού σε κάθε έκδοση και ένα γράφημα γραμμών (line chart) που απεικονίζει το ρυθμό ανάπτυξης των δομών δεδομένων (π.χ. Σχ. 4(β)) του λογισμικού σε κάθε έκδοση. Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν η εξέλιξη γίνεται με οργανωμένο συστηματικό τρόπο ο οποίος αντικατοπτρίζεται στην ύπαρξη επαναλαμβανόμενων μοτίβων στα ραβδογράμματα. Τα μοτίβα αυτά συνήθως έχουν την μορφή κυματισμών (spikes) οι κορυφές (peaks) των οποίων αντιστοιχούν σε εκδόσεις στις οποίες κυριαρχεί η θετική ανατροφοδότηση που οδηγεί στην αύξηση των λειτουργιών. Οι κοιλάδες (valleys) που ακολουθούν αντιστοιχούν σε εκδόσεις με μικρή, μηδενική ή μείωση των λειτουργιών. Στις εκδόσεις αυτές κυριαρχεί η αρνητική ανατροφοδότηση και οι αντίστοιχες δραστηριότητες συντήρησης. Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.



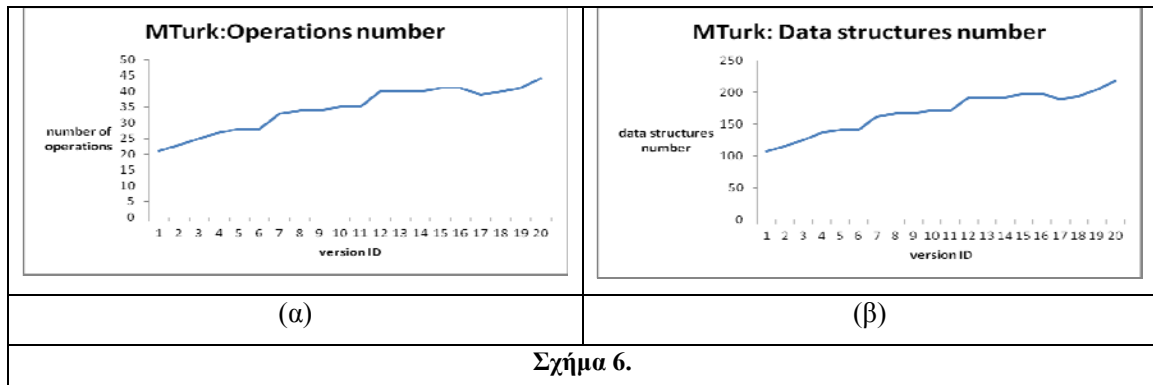
5. **Αποτίμηση του 4ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 4ου νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει τον ρυθμό εργασιών που σχετίζονται με τις λειτουργίες για κάθε έκδοση (π.χ. Σχ. 5(α)) και ένα γράφημα γραμμών (line chart) που απεικονίζει τον ρυθμό εργασιών που σχετίζονται με τις δομές δεδομένων του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 5(β)). Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν ο ρυθμός εργασιών και στις 2 περιπτώσεις τείνει να είναι σταθερός. Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.



SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

6. **Αποτίμηση του 5ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 5ου νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει το ρυθμό ανάπτυξης των λειτουργιών του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 4(α)) και ένα γράφημα γραμμών (line chart) που απεικονίζει το ρυθμό ανάπτυξης των δομών δεδομένων του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 4(β)). Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν και στα δυο γραφήματα εκδόσεις με μεγάλη αύξηση ακολουθούνται από εκδόσεις με μικρότερη, μηδενική ή αρνητική αύξηση. Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.
7. **Αποτίμηση του 6ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 6ου νόμου. Για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει το πλήθος λειτουργιών του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 6(α)) και ένα γράφημα γραμμών (line chart) που απεικονίζει το πλήθος των δομών δεδομένων του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 6(β)). Το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν και στα δυο γραφήματα παρατηρείται συνεχής αύξηση. Με βάση τις πληροφορίες αυτές και τις παραπάνω συνθήκες ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.

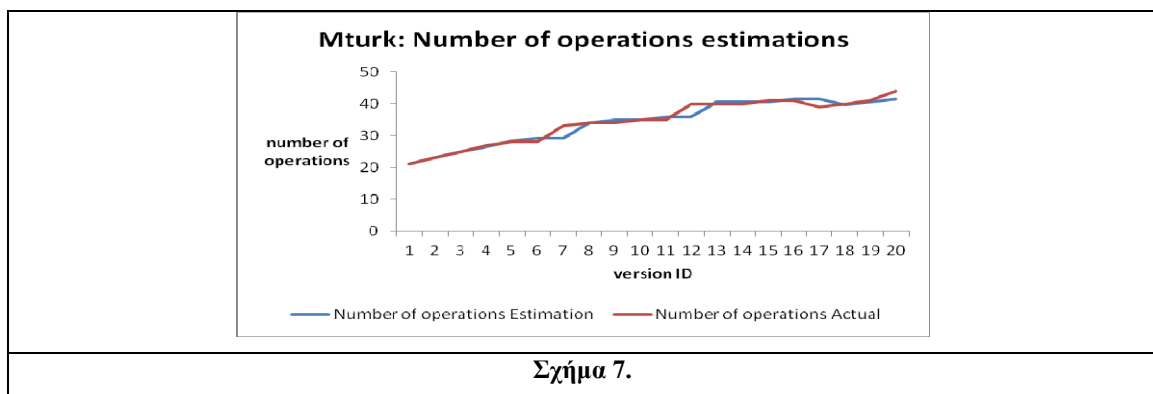
SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	



8. **Αποτίμηση του 7ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 7ου νόμου. Ο 7^{ος} νόμος είναι λογική συνέπεια του 2^{ου} και του 6^{ου} νόμου. Κατά συνέπεια αν έχει γίνει ήδη αποτίμηση των νόμων αυτών και ισχύουν το εργαλείο καταλήγει αυτόματα στο ότι ισχύει και ο 7^{ος} νόμος. Ο χρήστης στην περίπτωση αυτή μπορεί να προσθέσει ένα σύντομο σχολιασμό. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και ο σχολιασμός της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που δεν έχει γίνει αποτίμηση του 2^{ου} ή του 6^{ου} νόμου, το εργαλείο εμφανίζει σχετικό ενημερωτικό μήνυμα που παραπέμπει το χρήστη στην εκτέλεση αυτών των περιπτώσεων χρήσης. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.
9. **Αποτίμηση του 8ου νόμου:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία αποτίμησης του 8ου νόμου. Πιο συγκεκριμένα, το συμπέρασμα ότι ισχύει ο νόμος προκύπτει αν το πλήθος των λειτουργιών μιας μελλοντικής έκδοσης του συστήματος μπορεί να προβλεφτεί με ακρίβεια με βάση μια ανατροφοδοτούμενη φόρμουλα η οποία λαμβάνει υπόψη της το πλήθος των λειτουργιών του συστήματος σε προηγούμενες εκδόσεις. Η εν λόγω φόρμουλα που χρησιμοποιείται στη βιβλιογραφία είναι η εξής: $S_{i+1} = S_i + \frac{\bar{E}}{S_i^2}$. Στη φόρμουλα αυτή S_{i+1} είναι η πρόβλεψη του πλήθους των λειτουργιών μιας μελλοντικής έκδοσης, S_i είναι το πλήθος των λειτουργιών του συστήματος της τρέχουσας έκδοσης. Η μεταβλητή \bar{E} είναι μια εκτίμηση της προσπάθειας που απαιτείται για την υλοποίηση των αλλαγών στη μελλοντική έκδοση του λογισμικού. Η εκτίμηση αυτή υπολογίζεται ως η μέση τιμή της προσπάθειας που χρειάστηκε για την υλοποίηση των αλλαγών που έγιναν στις

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

προηγούμενες εκδόσεις του λογισμικού δηλ., $\bar{E} = \sum_{j=1}^i E_j / i$. Η τιμή E_j για μια προηγούμενη έκδοση υπολογίζεται με βάση την εξής φόρμουλα: $E_j = (S_j - S_1) / \sum_{k=1}^{j-1} \frac{1}{S_k^2}$. Επομένως, για την αποτίμηση το εργαλείο εμφανίζει στο χρήστη ένα γράφημα γραμμών (line chart) που απεικονίζει το εκτιμώμενο και το πραγματικό πλήθος λειτουργιών του λογισμικού σε κάθε έκδοση (π.χ. Σχ. 7). Με βάση τις πληροφορίες αυτές ο χρήστης επιλέγει το αν ισχύει ο νόμος ή όχι. Η επιλογή μπορεί να συνοδεύεται από ένα σύντομο σχολιασμό/αιτιολόγηση. Μετά την εκτέλεση των παραπάνω λειτουργιών η επιλογή καθώς και η αιτιολόγηση της διατηρούνται στην κύρια μνήμη μαζί με το ιστορικό εξέλιξης και εμφανίζονται στο χρήστη στην περίπτωση επανεκτίμησης του νόμου. Στην περίπτωση που ο χρήστης επιχειρήσει να εκτιμήσει την ισχύ του νόμου χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.



10. **Δημιουργία αναφοράς:** Ο χρήστης επιλέγει το ιστορικό ενός συστήματος λογισμικού και ακολούθως προχωρά στην διαδικασία μιας τελικής αναφοράς που περιλαμβάνει τις εκτιμήσεις του χρήστη σχετικά με την ισχύ των νόμων. Η αναφορά αποθηκεύεται σε ένα αρχείο κειμένου στο δίσκο. Στην περίπτωση που δεν έχει γίνει προηγουμένως αποτίμηση όλων των νόμων το εργαλείο εμφανίζει σχετικό ενημερωτικό μήνυμα που παραπέμπει το χρήστη στην εκτέλεση των περιπτώσεων χρήσης που απαιτούνται. Στην περίπτωση που ο χρήστης επιχειρήσει να δημιουργήσει μια τελική αναφορά χωρίς να επιλέξει το ιστορικό ενός συστήματος λογισμικού, το εργαλείο εμφανίζει μήνυμα λάθους.

4. Μη Λειτουργικές Απαιτήσεις

1. Όλες οι λειτουργίες του εργαλείου προσφέρονται μέσω κατάλληλης γραφικής επαφής με το χρήστη.

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

- Κατά την παράδοση του εργαλείου ο κατασκευαστής υποχρεούται να οργανώσει σεμινάριο επίδειξης των βασικών λειτουργιών του στους χρήστες.

5. Απαιτήσεις Τεκμηρίωσης

- Το εργαλείο θα παραδοθεί συνοδευόμενο από λεπτομερή οδηγό χρήσης.
- Το εργαλείο θα περιλαμβάνει και online help.

6. Απαιτήσεις Σχετικές με τη Διεργασία Ανάπτυξης

- Το εργαλείο θα βασιστεί στο **αντικειμενοστραφές στυλ**. Κατά συνέπεια η υλοποίηση του εργαλείου **θα αποτελείται από κλάσεις που αντιστοιχούν στο μοντέλο δεδομένων και σε όλες τις βασικές έννοιες που αφορούν στο πρόβλημα**.
- Οι μέθοδοι που θα χρησιμοποιηθούν για τον καθορισμό απαιτήσεων καθώς και τη σχεδίαση του συστήματος **θα βασίζονται στη UML**. Κατά συνέπεια θα χρησιμοποιηθεί αντικειμενοστρεφής προσέγγιση.
- Στη φάση της ανάλυσης των απαιτήσεων θα αναπτυχθεί ένα prototype** της γραφικής διεπαφής του εργαλείου με το χρήστη.

7. Εργαλεία υλοποίησης, βιβλιοθήκες κ.α.

Το εργαλείο θα υλοποιηθεί σε Java. Για την σχεδίαση και την υλοποίηση θα χρησιμοποιηθεί το εργαλείο Eclipse. Παρακάτω δίνονται χρήσιμα links για την εγκατάσταση της τελευταίας έκδοσης του Eclipse καθώς και των επεκτάσεων του (plug ins) που θα χρειαστούν για την εφαρμογή.

- Eclipse Kepler (<https://www.eclipse.org/downloads/packages/eclipse-standard-44-m5/lunam5>)
- Eclipse Papyrus plugin - για τη σχεδίαση UML διαγραμμάτων (<http://www.eclipse.org/papyrus/>)
- Tutorials για την εγκατάσταση του Papyrus και τη σχεδίαση διαγραμμάτων UML (http://www.eclipse.org/papyrus/usersTutorials/resources/TutorialOnPapyrusUSE_d20101001.pdf, <http://www.eclipse.org/papyrus/usersTutorials/usersTutorialsIndex.php>)
- Eclipse WindowBuilder plugin – προσφέρει ένα Eclipse editor για την υλοποίηση Java GUIs (<http://download.eclipse.org/windowbuilder/WB/release/R201309271200/4.3/>).
- Tutorial για τη χρήση του WindowBuilder (<http://www.java-forums.org/blogs/eclipse/795-using-windowbuilder-design-gui-eclipse.html>).
- JFreeChart βιβλιοθήκη για υλοποίηση γραφικών παραστάσεων σε Java (<http://www.jfree.org/jfreechart/>)

SoftwareEvolutionMonitor	Version: <1.0>
Περιγραφή Απαιτήσεων	Date: 11/09/2014
SoftwareEvolutionMonitor	

- Tutorial για το JFreeChart (<http://www.codeproject.com/Articles/650480/Introduction-to-JFreeChart>)