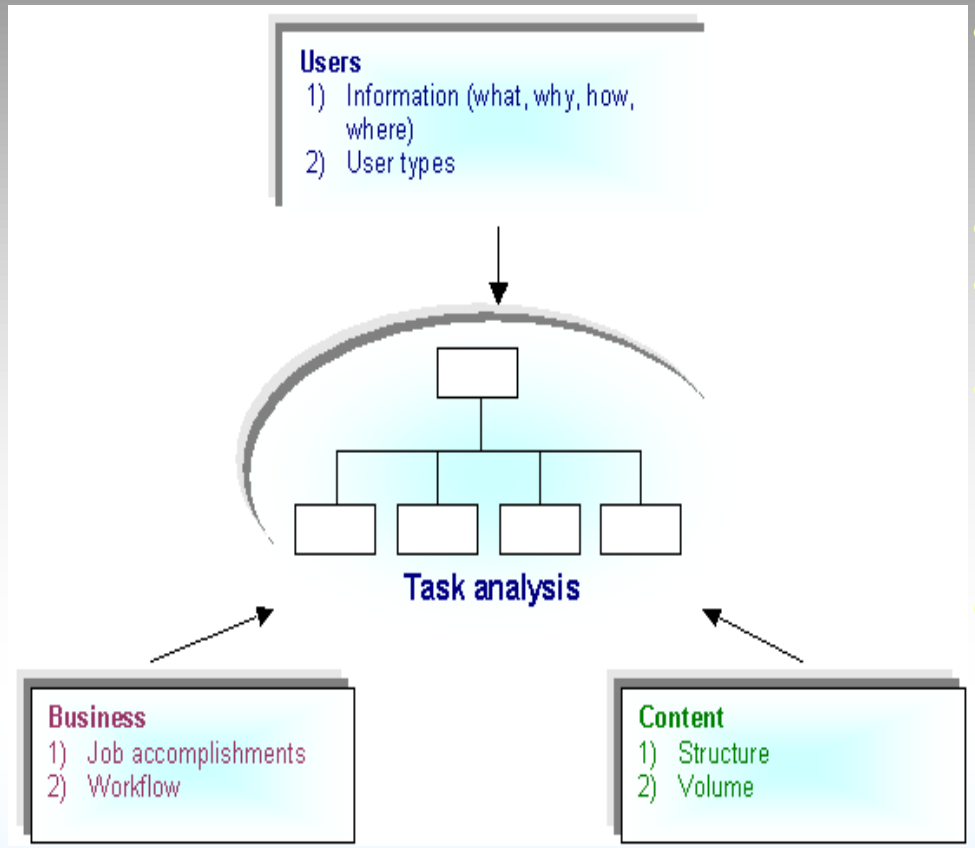
ΑΝΑΛΥΣΗ ΕΡΓΑΣΙΩΝ

(ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ ΔΙΕΠΑΦΗΣ ΜΕ ΤΟΥΣ ΧΡΗΣΤΕΣ)

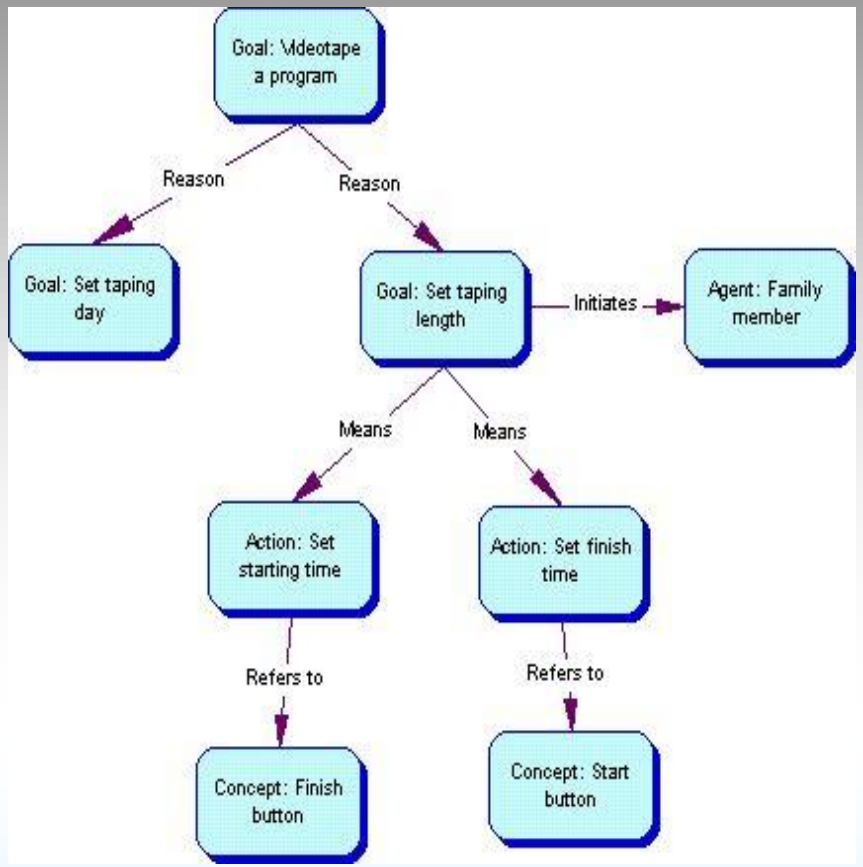
Ανάλυση Εργασιών (Task Analysis)



- Η **ανάλυση εργασιών** είναι η επεξεργασία της ανάλυσης του τρόπου που οι άνθρωποι κάνουν κάποιες δουλειές:
- τα πράγματα που κάνουν,
- τα πράγματα που πρέπει να γνωρίζουν,
- τα πράγματα στα οποία συμμετέχουν.

- Η ανάλυση εργασιών μοιάζει με τα γνωστικά μοντέλα χρηστών σε κάποια σημεία. Η διαφορά τους είναι ότι έχουν διαφορετικούς στόχους.

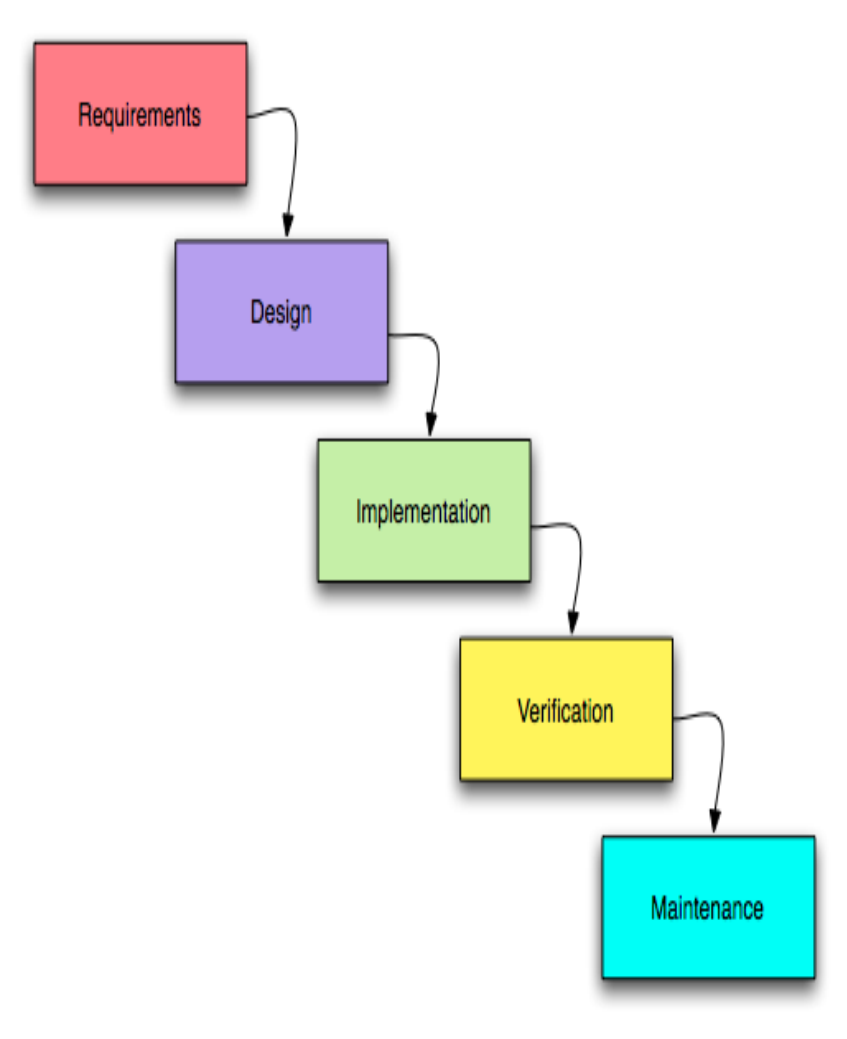
ΑΝΑΛΥΣΗ ΕΡΓΑΣΙΩΝ-ΣΤΟΧΟΙ ΧΡΗΣΤΩΝ



- Η Ανάλυση Εργασιών έχει στόχο την παρατήρηση του χρήστη από την εξωτερική σκοπιά, ενώ τα Μοντέλα Χρηστών έχουν στόχο την αναπαράσταση της κατάστασης της σκέψης του χρήστη.

- Δηλαδή στην Ανάλυση Εργασιών κανείς παρατηρεί το **τί γίνεται και όχι το γιατί**.
- Οπωσδήποτε όμως έχει άμεση σχέση με τους **στόχους του χρήστη**.

Ανάλυση Εργασιών (2)



- Με βάση τον κύκλο ζωής του λογισμικού ο οποίος αποτελείται από τις εξής φάσεις:
  1) Ανάλυση Απαιτήσεων
  2) Σχεδιασμός
  3) Υλοποίηση και επαλήθευση
  4) Ενσωμάτωση- Τεκμηρίωση
  5) Συντήρηση

- Η ανάλυση εργασιών ανήκει στη φάση της Ανάλυσης Απαιτήσεων ενώ τα γνωστικά Μοντέλα Χρηστών χρησιμοποιούνται κυρίως στο Σχεδιασμό και κατά τη φάση της Αξιολόγησης (προς το τέλος).

ΑΝΑΛΥΣΗ ΕΡΓΑΣΙΩΝ ΓΙΑ ΕΓΧΕΙΡΙΔΙΑ ΧΡΗΣΤΩΝ

- Η ανάλυση εργασιών είναι πολύ χρήσιμη και στον σχεδιασμό **εγχειριδίων χρηστών** στην Τεκμηρίωση του Λογισμικού.

- Στην Ανάλυση Εργασιών έχουμε και πάλι ως επί το πλείστον **διάσπαση εργασιών** σε κάποιες ιεραρχίες.

Διάσπαση εργασιών

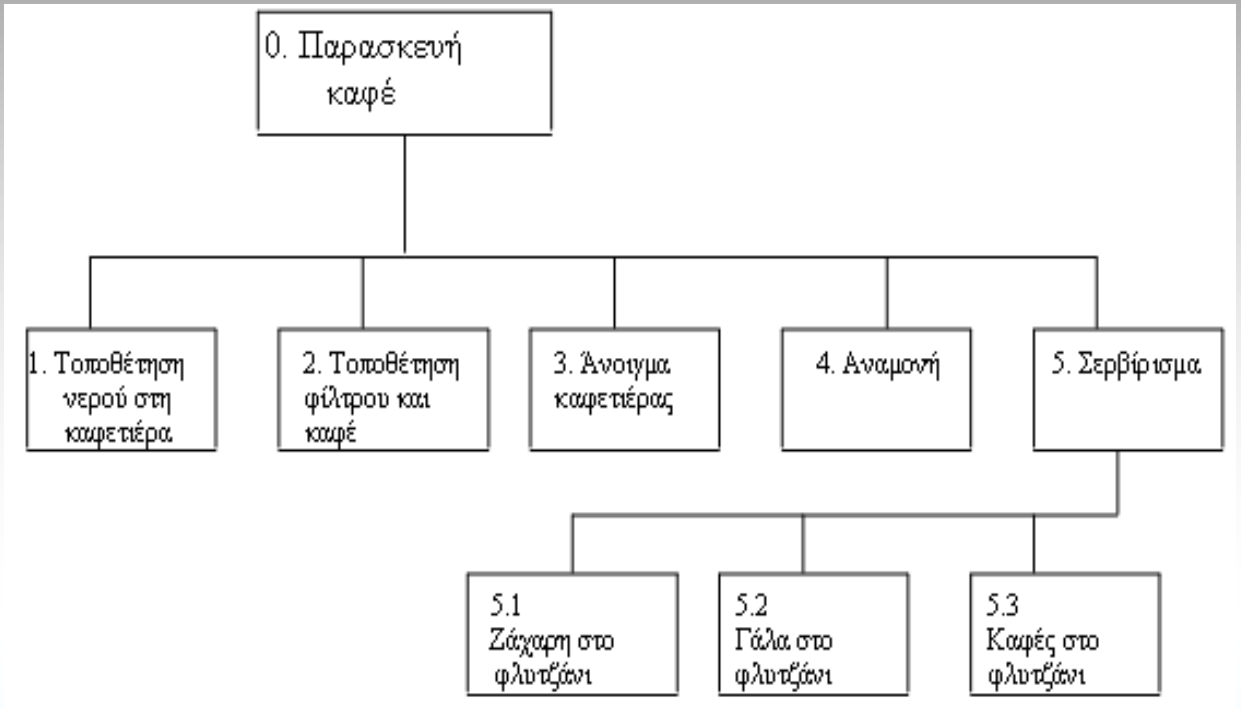
- Η Ιεραρχική Διάσπαση Εργασιών (Hierarchical Task Decomposition) περιλαμβάνει ιεραρχίες εργασιών και υπο- εργασιών
- καθώς και
- **σχέδια** (πλάνα) που περιγράφουν σε ποια σειρά πρέπει να γίνουν κάποια πράγματα.
- Μπορούμε να παραστήσουμε τις ιεραρχίες με κείμενο ή με διαγράμματα.

- Για παράδειγμα η **εργασία του καθαρίσματος του σπιτιού** μπορεί να αναλυθεί ως εξής:

0. Για να καθαρίσουμε το σπίτι
1. Βγάζουμε την ηλεκτρική σκούπα
2. Τη βάζουμε στην πρίζα
3. Καθαρίζουμε τα δωμάτια
    3.1 Πρώτο δωμάτιο
    3.2 Δεύτερο δωμάτιο
    3.3 Τρίτο δωμάτιο
4. Αδειάζουμε τη σακκούλα σκόνης
5. Βάζουμε την ηλεκτρική σκούπα στη θέση της.

Παράδειγμα διάσπασης εργασιών

- Ένα άλλο παράδειγμα σε διαγραμματική αναπαράσταση : **η εργασία του να φτιάξουμε καφέ φίλτρου**.



- Αυτό το διάγραμμα μπορεί να εμπλουτιστεί με εργασίες που έχουν παραλειφθεί. Λάθη και παραλείψεις μπορούν να εντοπισθούν με τη συζήτηση με κάποιον εμπειρογνώμονα στο Πεδίο (Domain Expert).

Χρήσεις της ανάλυσης εργασιών

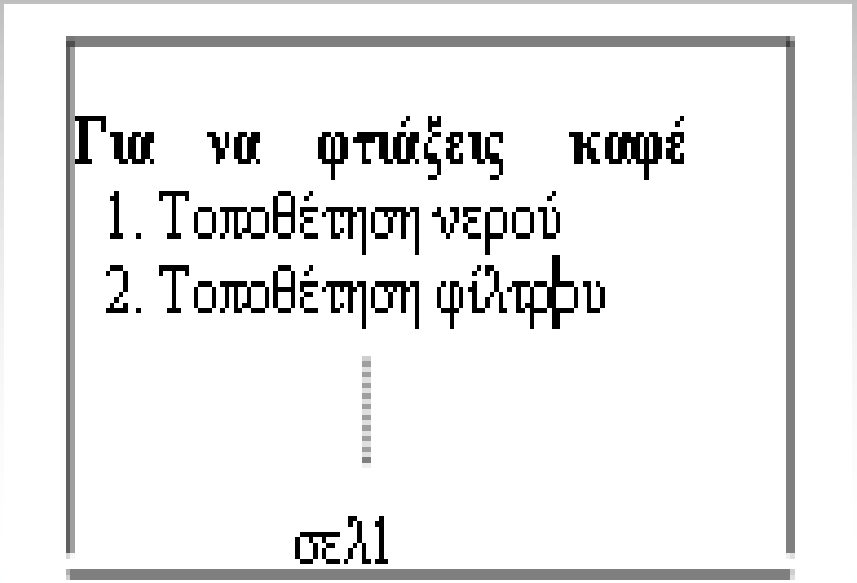
- Η ανάλυση εργασιών είναι πολύ χρήσιμη για τα εξής:

1. Εγχειρίδια χρήστη και διδασκαλία

2. Ανάλυση απαιτήσεων

3. Λεπτομερειακό σχεδιασμό συστήματος διεπαφής

ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ ΜΕ ΙΕΡΑΡΧΙΚΗ ΔΟΜΗ



Η ιεραρχική δομή της HTA χρησιμοποιείται για τη δόμηση των εγχειριδίων χρήστη (user manuals).

Για παράδειγμα, αν θέλουμε να γράψουμε ένα εγχειρίδιο χρήστη για το πώς να φτιάχνει καφέ βασιζόμαστε στην ιεραρχική ανάλυση εργασιών (HTA).

Βασίζουμε κάθε σελίδα σε ένα επίπεδο της διάσπασης των εργασιών και του πλάνου που σχετίζεται.

- Κάθε εργασία που αναλύεται περισσότερο θα πρέπει να έχει την αναφορά της σε άλλη σελίδα.

Χρήσεις της ανάλυσης εργασιών (2)

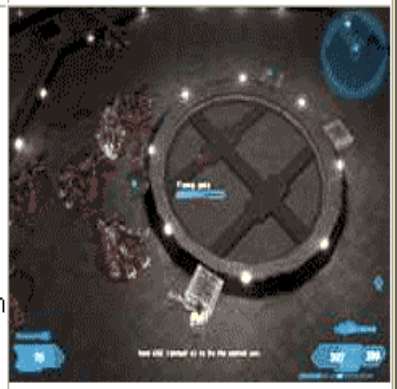
2) Στην Ανάλυση Απαιτήσεων μπορεί να χρησιμοποιηθεί για να δούμε τι απ' όλα όσα αναφέρονται στην Ανάλυση Εργασιών έχουν άμεση σχέση με το σύστημα ώστε να προκύψει η Ανάλυση Απαιτήσεων

(π.χ. το ότι βάζουμε ζάχαρη στα φλυτζάνια δεν έχει άμεση σχέση με το μηχανισμό της καφετιέρας).

Σχεδιασμός συστήματος διεπαφής με το χρήστη

- Ο κύριος στόχος του σχεδιασμού ενός συστήματος διεπαφής είναι η μέγιστη δυνατή χρησιμοποιησιμότητα (maximum usability). Ο σχεδιαστής επομένως έχει να απαντήσει δύο καίρια ερωτήματα :
  1) Πώς μπορεί να αναπτυχθεί ένα αλληλεπιδρών σύστημα ώστε να έχει μέγιστη χρησιμοποιησιμότητα;
  2) Πώς μπορεί η χρησιμοποιησιμότητα να μετρηθεί;

Σχεδιασμός συστήματος διεπαφής με το χρήστη (2)

Problem	Rating
When fixing an item, the feedback is displayed too far away from the location being fixed.	Intermediate
Description The player often fixes items in the game. Fixing an item is done by moving close to the item and holding down E. A progress bar is displayed and the item is fixed when the bar reaches the end. The progress bar is displayed too far away from the item fixed. Presenting the feedback closer to the location at which the action takes place makes it easier to understand that the feedback and action are linked to each other. It also reduces the need for the user to move his/her attention between the feedback and the action.	
Solution Display the progress bar close to the location it refers to.	
Developers' comment This seems like a no-brainer but it went unnoticed for a long time by the development team, because it had grown so accustomed to the game and its features. The feature had only a vague description in the design document (loosely translated to "a repair bar is displayed on the screen"). Had the document been given more thought or had it been read by usability experts before implementing the feature, the feature might have been done correctly from the get-go.	

- Υπάρχουν δύο τρόποι να απαντήσει κανείς αυτές τις ερωτήσεις:

α) **Με παραδείγματα**. Σ' αυτή την περίπτωση κάποια συστήματα διεπαφής τα οποία είναι κοινώς αποδεκτό ότι είναι επιτυχημένα μπορούν να χρησιμοποιηθούν σαν παραδείγματα για σχεδιασμό μελλοντικών συστημάτων.

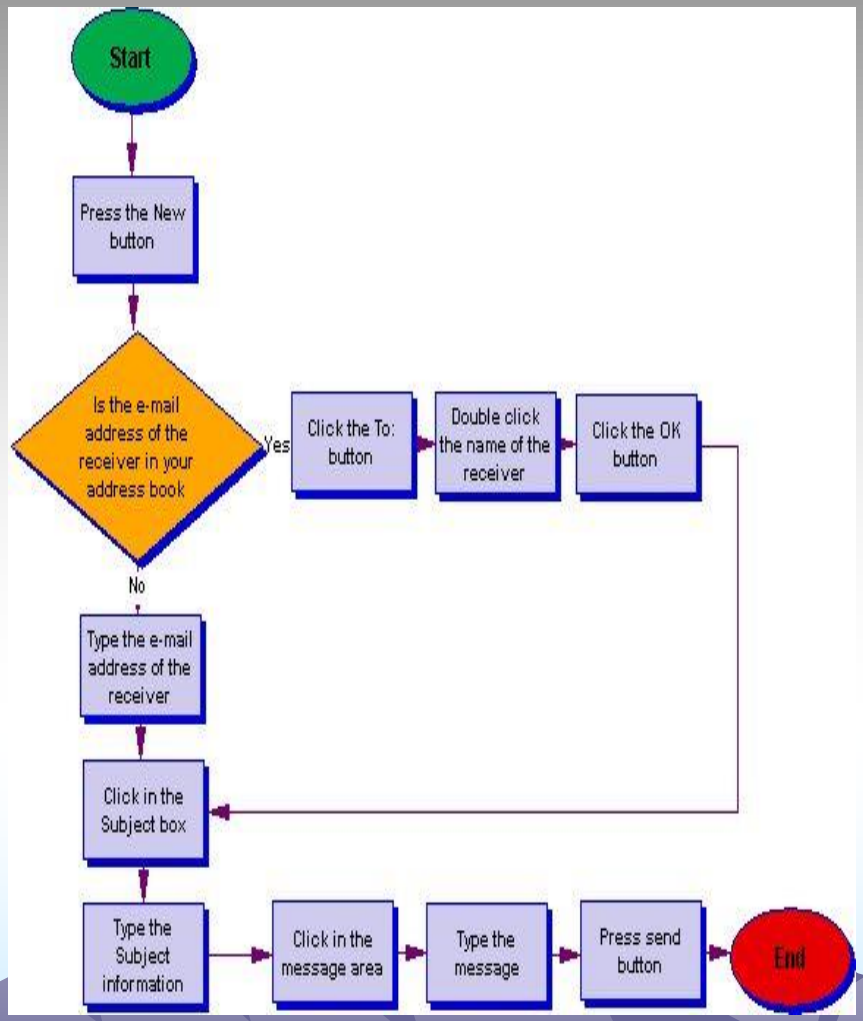
β) **Με κάποιες γενικές αρχές για αποτελεσματική αλληλεπίδραση.** Αυτός ο τρόπος είναι πιο θεωρητικός. Κάποιες γενικές αρχές που προέρχονται από γνώση πάνω στην ψυχολογική, κοινωνική και υπολογιστική πλευρά ενός χώρου- προβλήματος (problem domain) και οι οποίες καθοδηγούν το σχεδιασμό.

ΓΕΝΙΚΕΣ ΑΡΧΕΣ ΕΥΧΡΗΣΤΙΑΣ

- Εδώ θα ασχοληθούμε με τις γενικές αρχές που θα μπορούσαν να υποστηρίξουν την **ευχρηστία.**

- Βέβαια δεν μπορούμε να ισχυριστούμε ότι υπάρχει ένας ακριβής κατάλογος γενικών αρχών αλλά οπωσδήποτε κάποια αναφορά σε ορισμένες αρχές παρέχει κάποια βοήθεια στο σχεδιαστή.

Αρχές που υποστηρίζουν την *ευχρηστία*



- Θα μπορούσαμε να διαιρέσουμε τις γενικές αρχές σε τρείς μεγάλες κατηγορίες ανάλογα με τον τομέα στον οποίο αναφέρονται. Οι τομείς είναι οι εξής:

1) **Ευκολία εκμάθησης (Learnability)**
Δηλαδή πόσο εύκολο είναι για τον χρήστη να μάθει να χρησιμοποιεί το σύστημα διεπαφής στο μέγιστο βαθμό.

2) **Ευκαμψία (Flexibility)**
Δηλαδή κατά πόσο υποστηρίζονται πολλοί τρόποι επικοινωνίας του χρήστη με το σύστημα.

3) **Ανθεκτικότητα (Robustness)**
Δηλαδή το επίπεδο της υποστήριξης που παρέχεται από το σύστημα στα πιθανά λάθη που μπορεί να κάνει ο χρήστης.

Ευκολία εκμάθησης, Ευκαμψία, Ανθεκτικότητα

- Αυτές οι τρείς έννοιες μπορούν να αναλυθούν με βάση άλλες σημαντικές έννοιες που σχετίζονται με την ευχρηστία.

ΑΝΑΛΥΣΗ
ΒΑΣΙΚΩΝ ΑΡΧΩΝ ΠΟΥ ΥΠΟΣΤΗΡΙΖΟΥΝ ΤΗΝ *ΕΥΧΡΗΣΤΙΑ*

- Η **Ευχρηστία** υποστηρίζεται από:

1. Ευκολία στην εκμάθηση (Learnability)

η οποία υποστηρίζεται από:

i) Προβλεψιμότητα
ii) Σύνθεση
iii) Οικειότητα
iv) Γενίκευση
v) Συνέπεια

2. Ευκαμψία (Flexibility)

η οποία υποστηρίζεται από:

i) Δυνατότητα αλλαγής εργασιών
ii) Πολλαπλός έλεγχος
iii) Προσαρμοσιμότητα

3. Ανθεκτικότητα (Robustness)

η οποία υποστηρίζεται από:

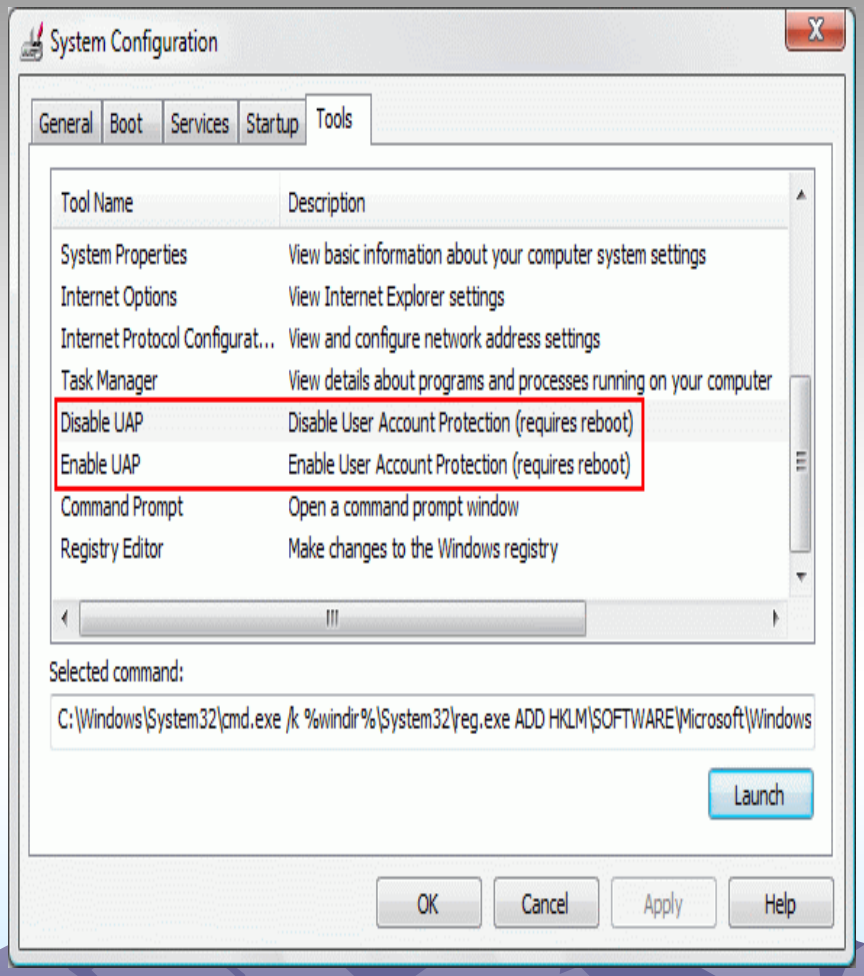
i) Δυνατότητα παρατήρησης
ii) Ανανηψιμότητα
iii) Λειτουργικότητα

1. Ευκολία στην εκμάθηση
- i) *Προβλεψιμότητητα*

- Η ευκολία στην εκμάθηση υποστηρίζεται από τις εξής βασικές γενικές αρχές:

1) **Προβλεψιμότητα**:
- Δηλαδή υποστήριξη προς το χρήστη ώστε να μπορεί εύκολα να καθορίσει το αποτέλεσμα μιας μελλοντικής ενέργειας βασισμένος (ο χρήστης) σε εμπειρία παλιότερης επικοινωνίας με το σύστημα.

ΠΡΟΒΛΕΨΙΜΟΤΗΤΑ



- Η προβλεψιμότητα έχει να κάνει με την ικανότητα του χρήστη να καθορίσει τα **αποτελέσματα κάποιων λειτουργιών** (π.χ. τι θα γίνει αν πατήσει το ποντίκι σε μια συγκεκριμένη θέση).
- Επίσης έχει να κάνει με την ικανότητα του να καταλάβει **ποιες λειτουργίες επιτρέπονται**.
- Μια **γενική αρχή** που σχετίζεται με το ποιες λειτουργίες επιτρέπονται είναι η **ορατότητα των λειτουργιών** δηλαδή κατά πόσο υπάρχουν ενδείξεις στο χρήστη για το ποιές λειτουργίες μπορεί να εκτελέσει σε κάποια δεδομένη στιγμή.

1. Ευκολία στην εκμάθηση– ii) *Σύνθεση*

Σύνθεση:

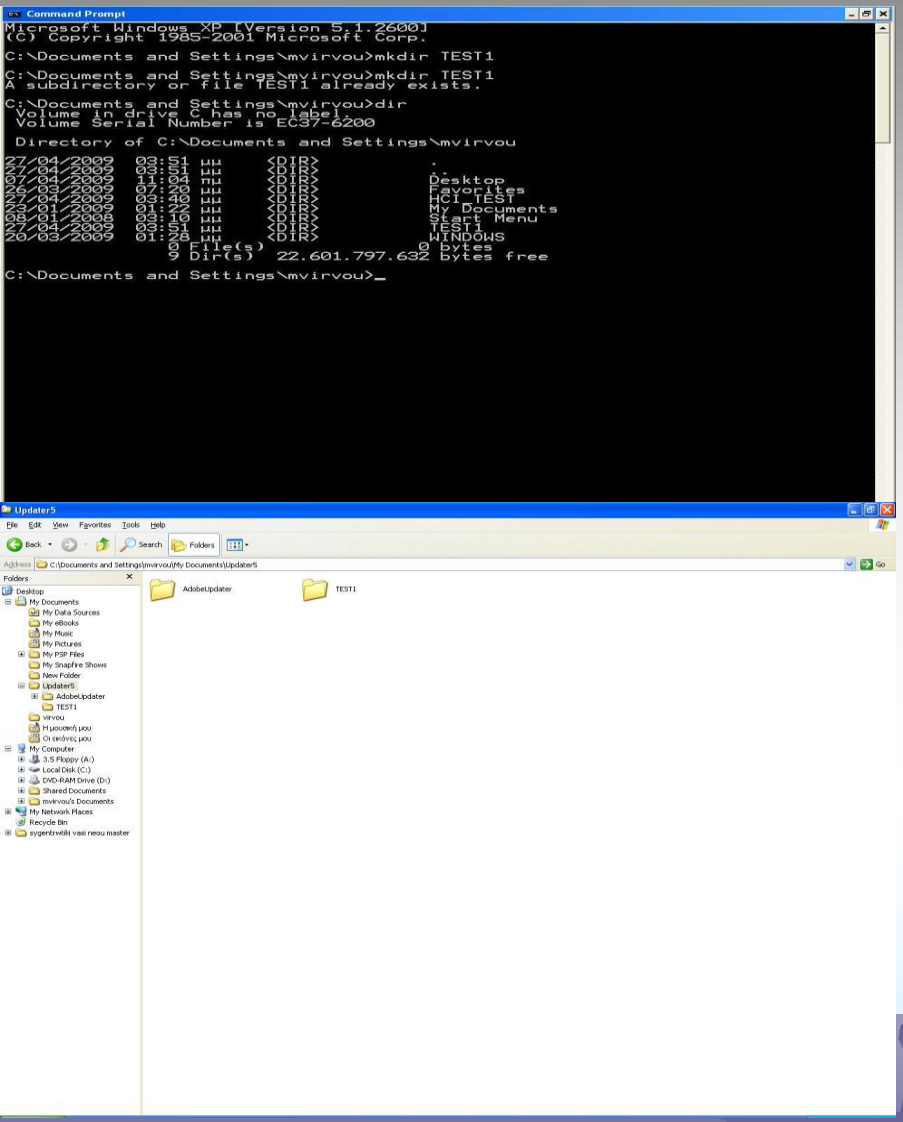
- ***Δηλαδή κατά πόσο ο χρήστης μπορεί να φτιάξει ένα μοντέλο της τρέχουσας κατάστασης του συστήματος βασισμένος στις ενέργειες που έχει κάνει.***

1. Ευκολία στην εκμάθηση– ii) *Σύνθεση*

Σύνθεση:

- ***Δηλαδή κατά πόσο ο χρήστης μπορεί να φτιάξει ένα μοντέλο της τρέχουσας κατάστασης του συστήματος βασισμένος στις ενέργειες που έχει κάνει.***
- Π.χ Στην κλήση ασανσέρ: Εάν κανένα κουμπάκι δεν ανάψει, τότε ο χρήστης μπορεί να νομίζει ότι δεν δόθηκε σωστά η εντολή του.
- Σε αυτήν την περίπτωση μάλλον θα ξαναδώσει την εντολή, θα προσπαθήσει να αξιολογήσει την ενέργειά του με άλλο τρόπο
- ή θα αλλάξει πλάνο.

ΣΥΝΘΕΣΗ



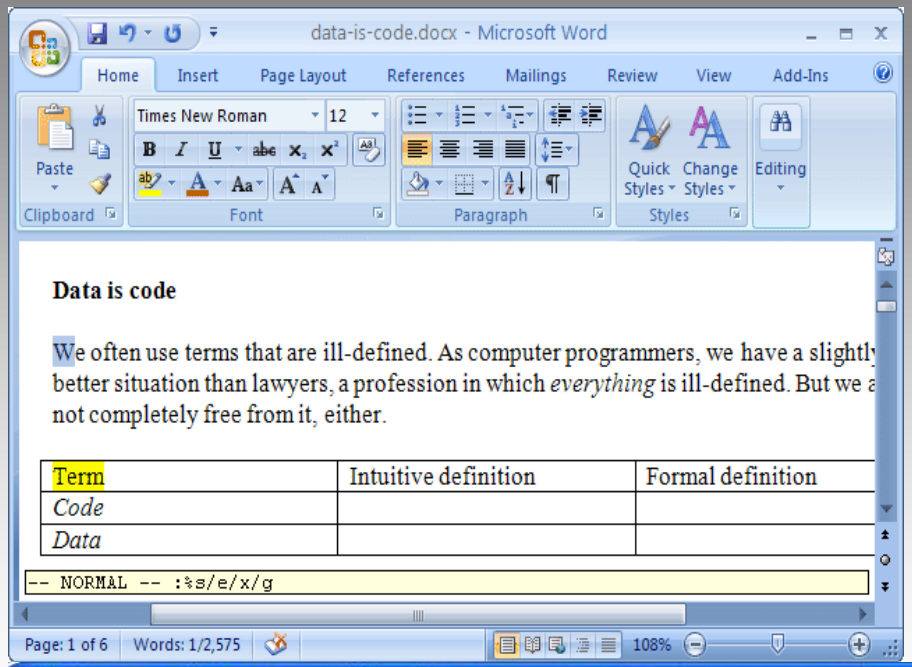
- **Σύνθεση είναι η δυνατότητα του χρήστη**
- **να εκτιμήσει τα αποτελέσματα των ενεργειών που έχει κάνει στην κατάσταση του συστήματος.**
- **Όταν μια λειτουργία αλλάζει κάτι από το σύστημα, είναι σημαντικό η αλλαγή αυτή να επιδεικνύεται στο χρήστη.**
- ***Για παράδειγμα σε ένα σύστημα διεπαφής με βάση***
- ***γλώσσα εντολών όταν μετακινούμε ένα αρχείο από έναν κατάλογο σε κάποιον άλλο η αλλαγή δεν φαίνεται άμεσα***
- ***ενώ όταν κάνουμε την ίδια ενέργεια με το ποντίκι σε ένα γραφικό περιβάλλον, η αλλαγή φαίνεται αμέσως.***

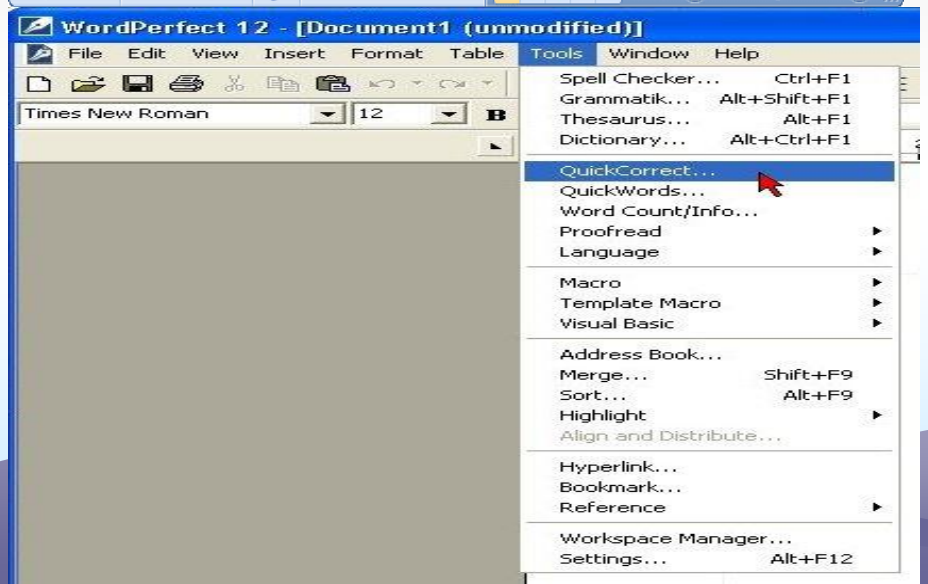
1. Ευκολία στην εκμάθηση– iii) *Οικειότητα*

3) **Οικειότητα:**

- Ο βαθμός στον οποίο
- η γνώση και εμπειρία του χρήστη *από άλλα πεδία εφαρμογών* σε πληροφοριακά συστήματα ή άλλα συστήματα
- *μπορεί να χρησιμοποιηθεί στην επικοινωνία με το καινούριο σύστημα*.

1. Ευκολία στην εκμάθηση – iii) Οικειότητα



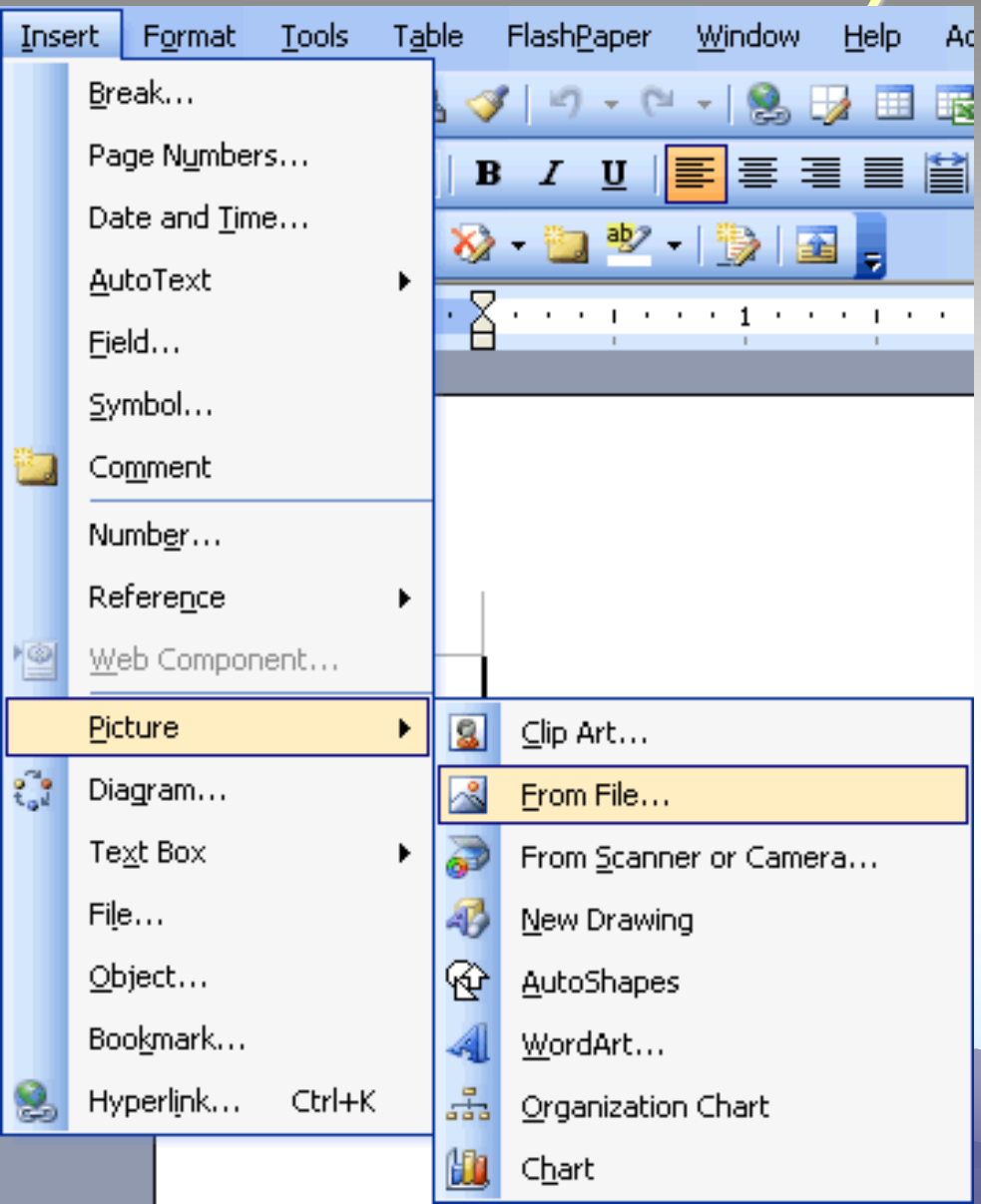


Για παράδειγμα, κατά πόσο η εξοικείωση με ένα πρόγραμμα κειμενογράφου (π.χ *Word*)

μπορεί να βοηθήσει στη χρήση άλλου κειμενογράφου (π.χ. *Word Perfect*)

Αυτό θα μπορούσε να συμβεί αν τα εικονίδια στη γραμμή εργαλείων (toolbar) είναι ίδια σε μεγάλο βαθμό κ.λπ.

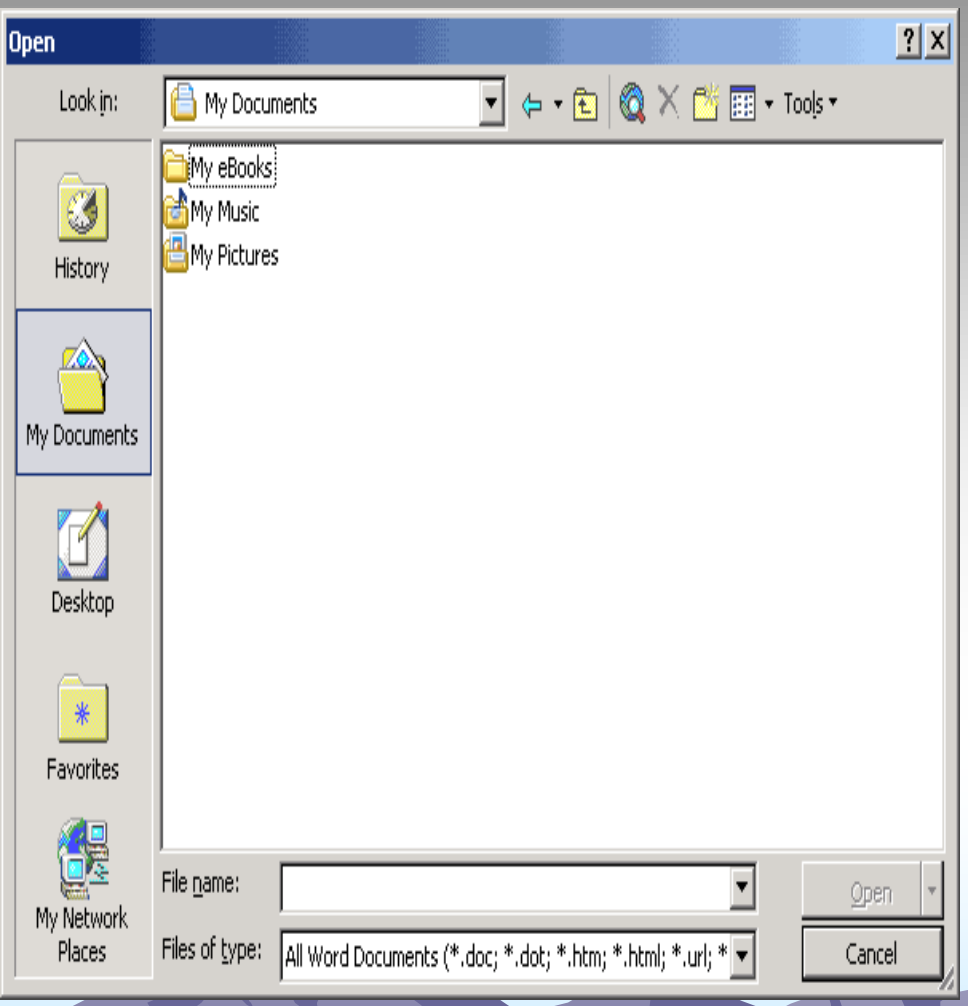
1. Ευκολία στην εκμάθηση– iv) *Γενίκευση*



4) **Γενίκευση:**

- Δηλαδή κατά πόσο ο χρήστης μπορεί να προβλέψει τις ενέργειες σε παρόμοιες καταστάσεις.
- Π.χ. Η εντολή insert σε διάφορα πλαίσια

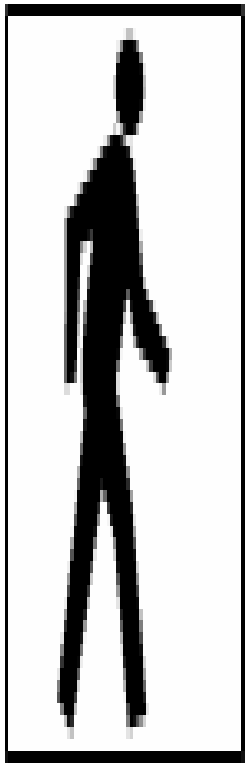
1. Ευκολία στην εκμάθηση – ν) *Συνέπεια*



5) **Συνέπεια :**

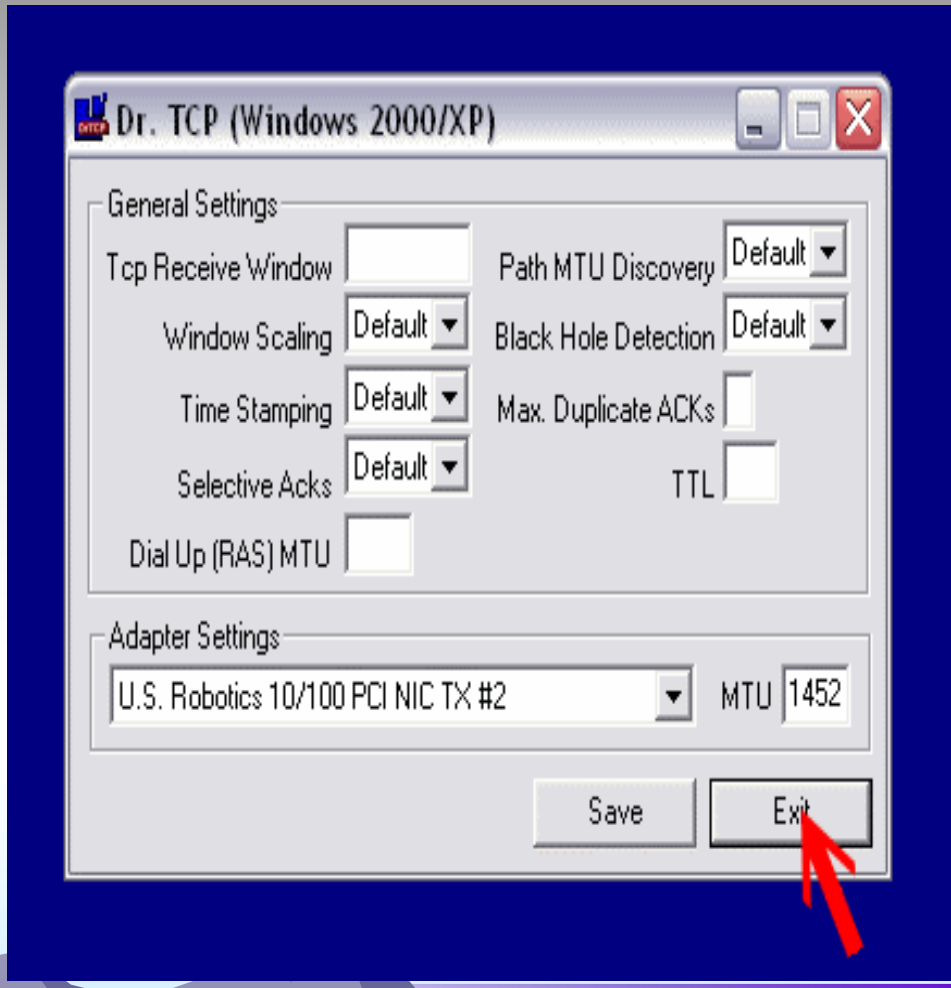
- Δηλαδή κατά πόσο χρησιμοποιούνται οι ίδιες συμβάσεις καθ' όλη τη διάρκεια του προγράμματος

ΣΥΝΕΠΕΙΑ- Παράδειγμα

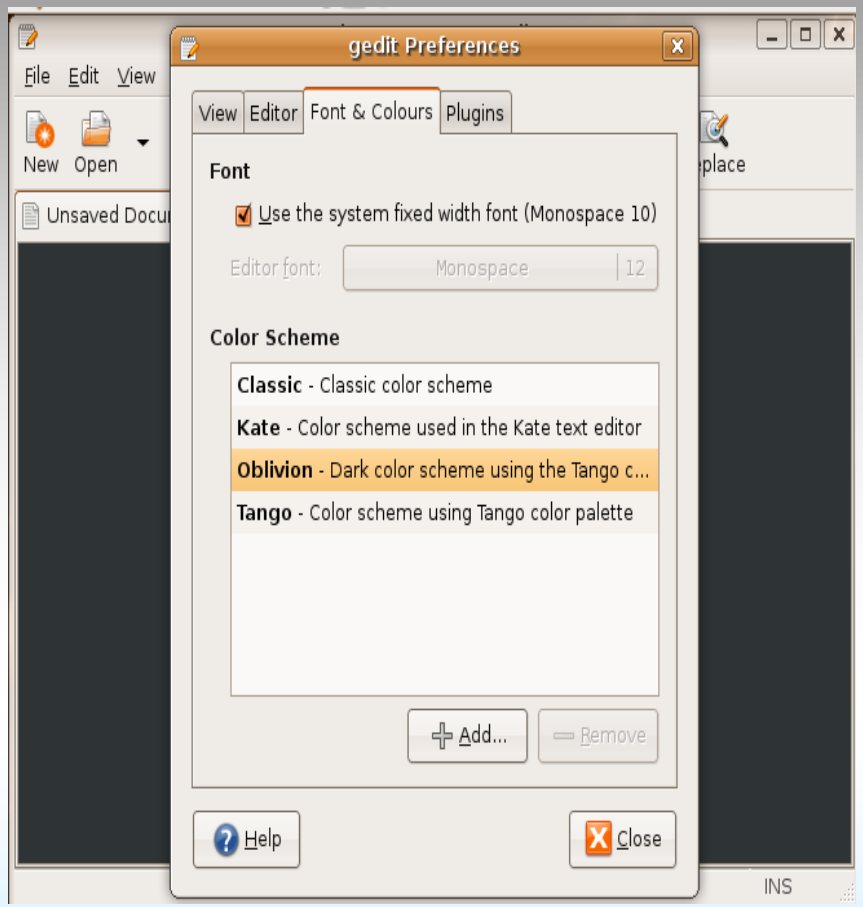


- Για παράδειγμα, αν στην πρώτη οθόνη, το πρώτο κουμπί εξόδου από το πρόγραμμα γράφει EXIT και βρίσκεται στην κάτω αριστερή άκρη της οθόνης, τότε θα πρέπει να παραμείνει έτσι σε όλες τις οθόνες του προγράμματος.
- Δεν θα ήταν σκόπιμο να αλλάζει θέση στην οθόνη ή να αλλάξει το κουμπί το ίδιο (π.χ. με εικονίδιο το οποίο να δείχνει ένα ανθρωπάκι να φεύγει, όπως στο παρακάτω σχήμα).
- **Ασυνέπεια**: Διαφορετικές αναπαραστάσεις για το ίδιο πράγμα

ΣΥΝΕΠΕΙΑ: ΤΑ ΚΟΥΜΠΙΑ ΝΑ ΠΑΡΑΜΕΝΟΥΝ ΙΔΙΑ ΣΕ ΟΛΗ ΤΗΝ ΕΦΑΡΜΟΓΗ
(π.χ. Για να εγκαταλείψουμε την εφαρμογή...)

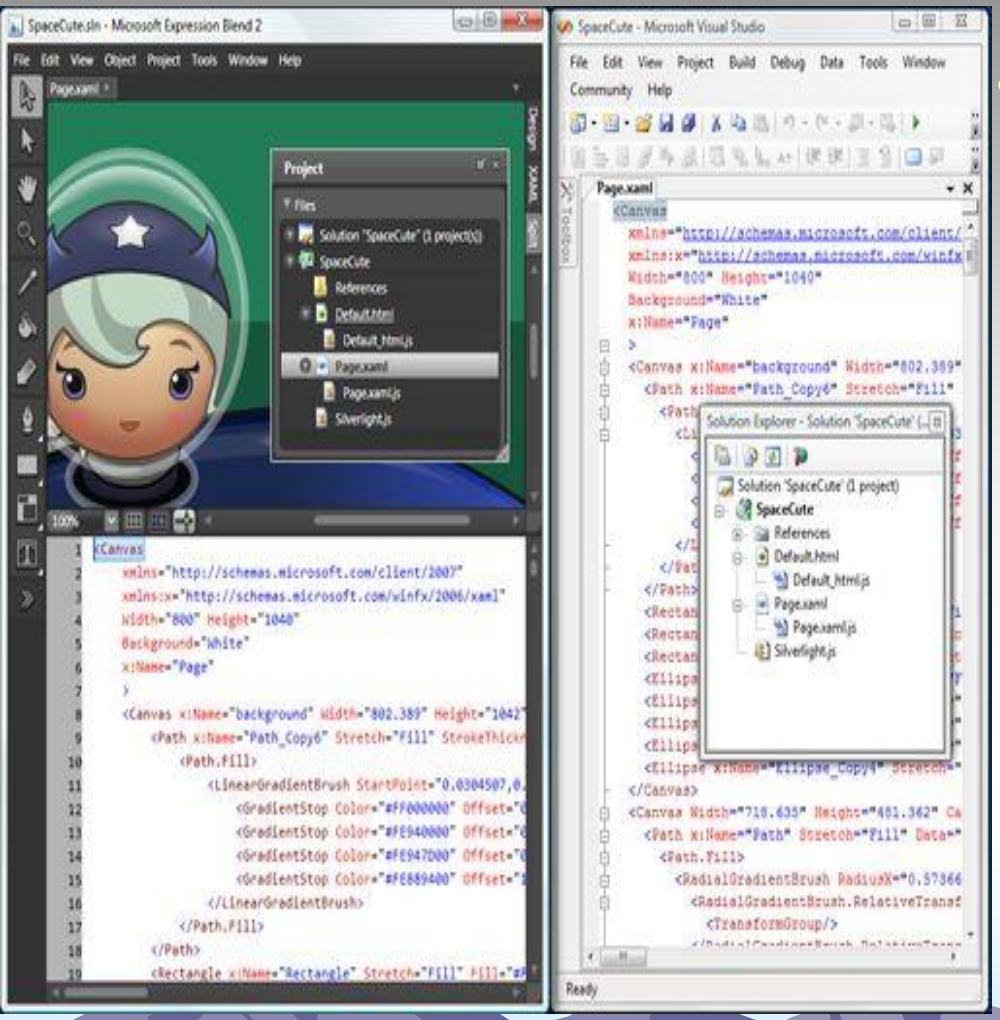


ΣΥΝΕΠΕΙΑ: ΤΑ ΧΡΩΜΑΤΑ ΚΑΙ Η ΔΟΜΗ ΝΑ ΠΑΡΑΜΕΝΟΥΝ ΙΔΙΑ ΣΕ ΟΛΗ ΤΗΝ ΕΦΑΡΜΟΓΗ



Ο ΤΡΟΠΟΣ ΠΟΥ ΠΑΙΡΝΕΙ ΒΟΗΘΕΙΑ Ο ΧΡΗΣΤΗΣ ΝΑ ΠΑΡΑΜΕΝΕΙ ΙΔΙΟΣ ΣΕ ΟΛΗ ΤΗΝ ΕΦΑΡΜΟΓΗ

2. Ευκαμψία του συστήματος διεπαφής: i) Πολλαπλός έλεγχος



- Η ευκαμψία του συστήματος διεπαφής εξαρτάται πολύ από τους παρακάτω παράγοντες:

i) **Πολλαπλό έλεγχο:** Δηλαδή κατά πόσο ο χρήστης έχει δυνατότητα διαλόγου για περισσότερες από μία εργασίες τη φορά.

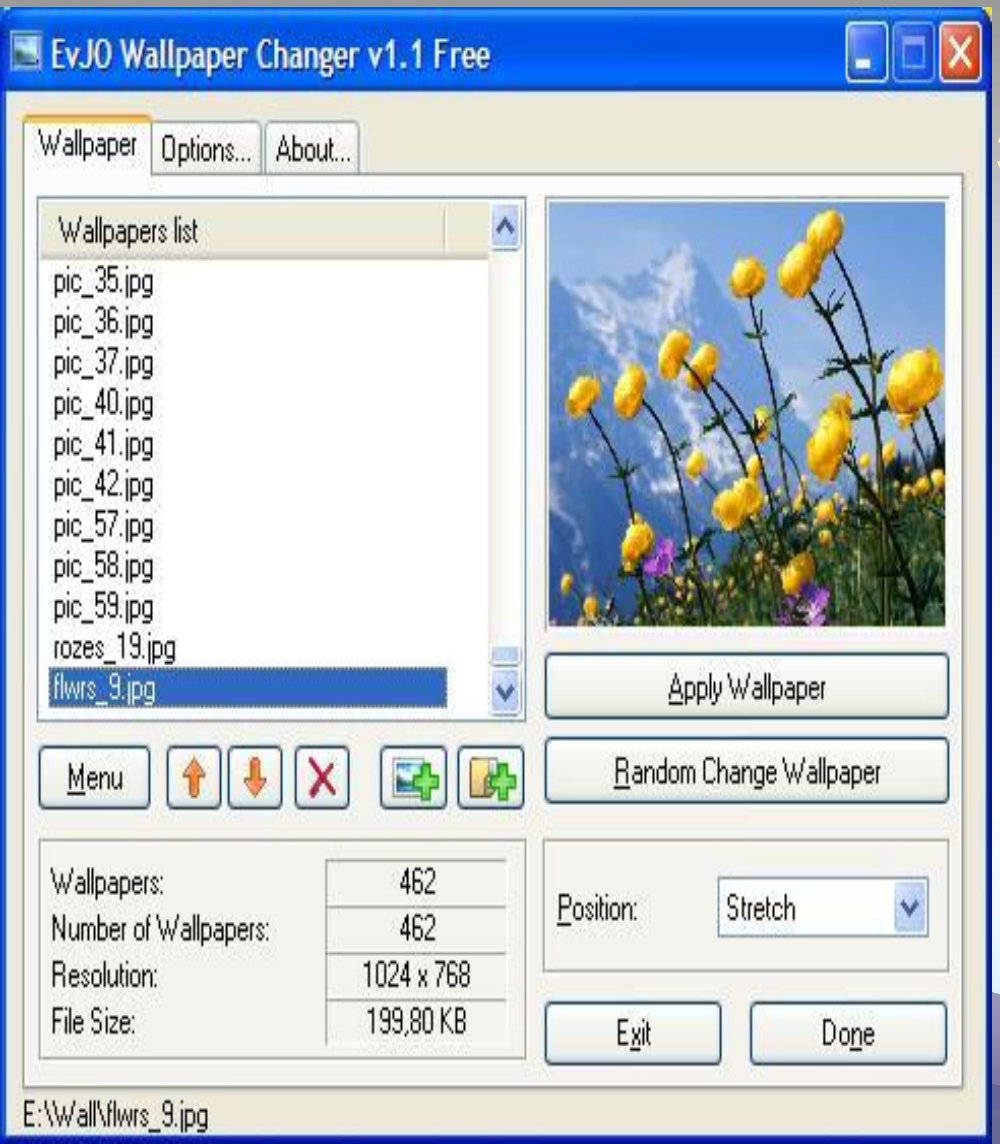
Ευκαμψία του συστήματος διεπαφής: ii) Δυνατότητα αλλαγής εργασιών

ii) Δυνατότητα αλλαγής εργασιών:

Δηλαδή κατά πόσο μπορεί ο χρήστης

να περάσει εύκολα από μία εργασία σε μία άλλη.

Ευκαμψία του συστήματος διεπαφής:
iii) Προσαρμοσιμότητα

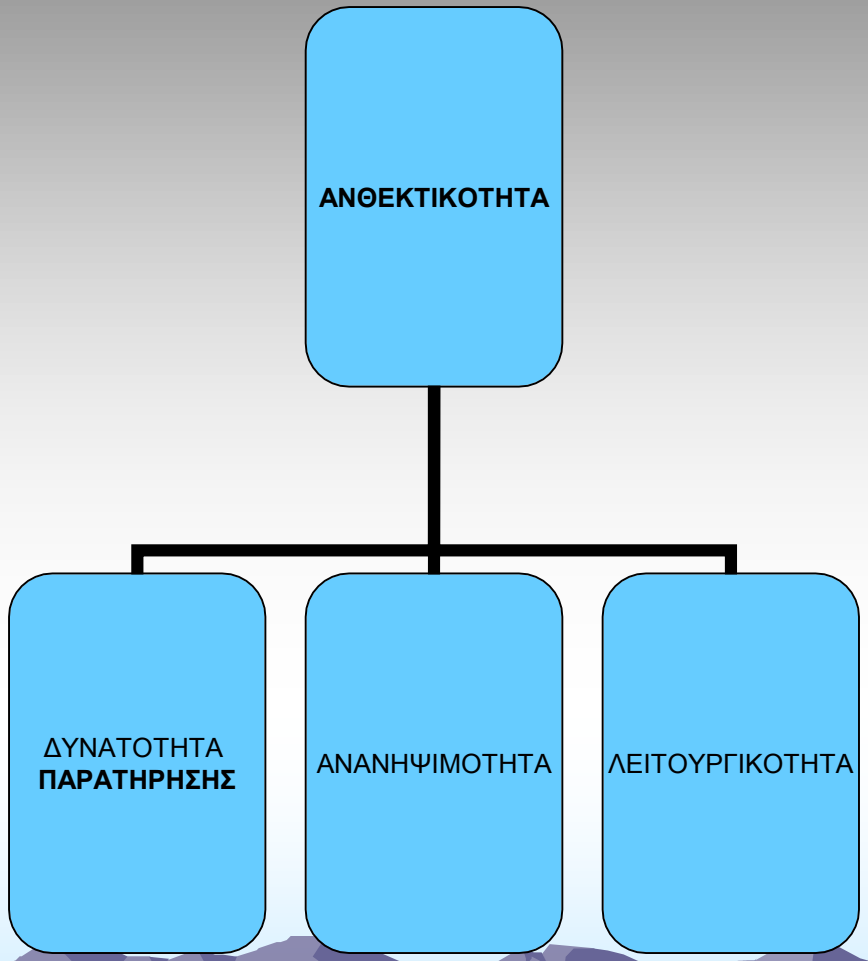


3) **Προσαρμοσιμότητα:**

Η προσαρμοσιμότητα εξαρτάται από τη δυνατότητα αλλαγής παραμέτρων ώστε να ταιριάζουν στο χρήστη.

Για παράδειγμα, αν ένας χρήστης μπορεί να αλλάξει το μέγεθος των γραμμάτων στα κουτιά διαλόγου ή το χρώμα του υποβάθρου ή το Wallpaper κ.λπ.

Ανθεκτικότητα του συστήματος διεπαφής



- Η ανθεκτικότητα (robustness) εξαρτάται πολύ από τους παρακάτω παράγοντες:

  1) **Δυνατότητα παρατήρησης**: Κατά πόσο βλέπει ο χρήστης τα αποτελέσματα κάποιων ενεργειών.

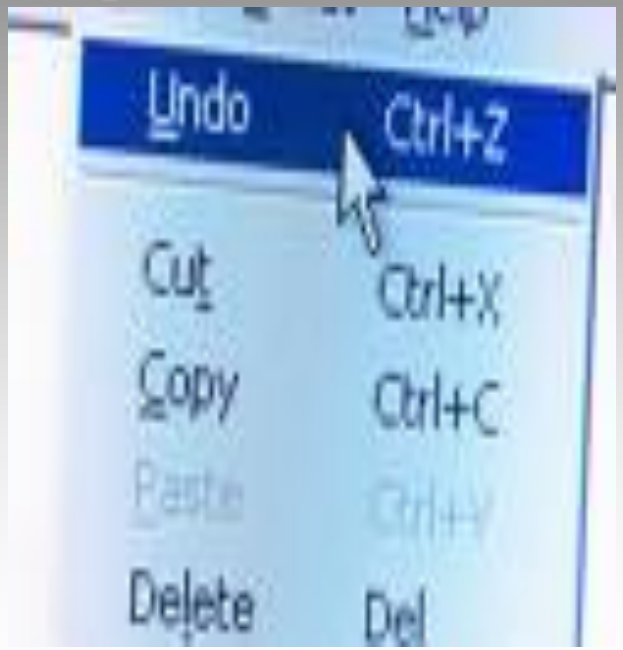
  2) **Ανανηψιμότητα**: Δυνατότητα επανόρθωσης λάθους.

  3) **Λειτουργικότητα**: Κατά πόσο υποστηρίζονται όλες οι ενέργειες που θα ήθελε να κάνει ο χρήστης.

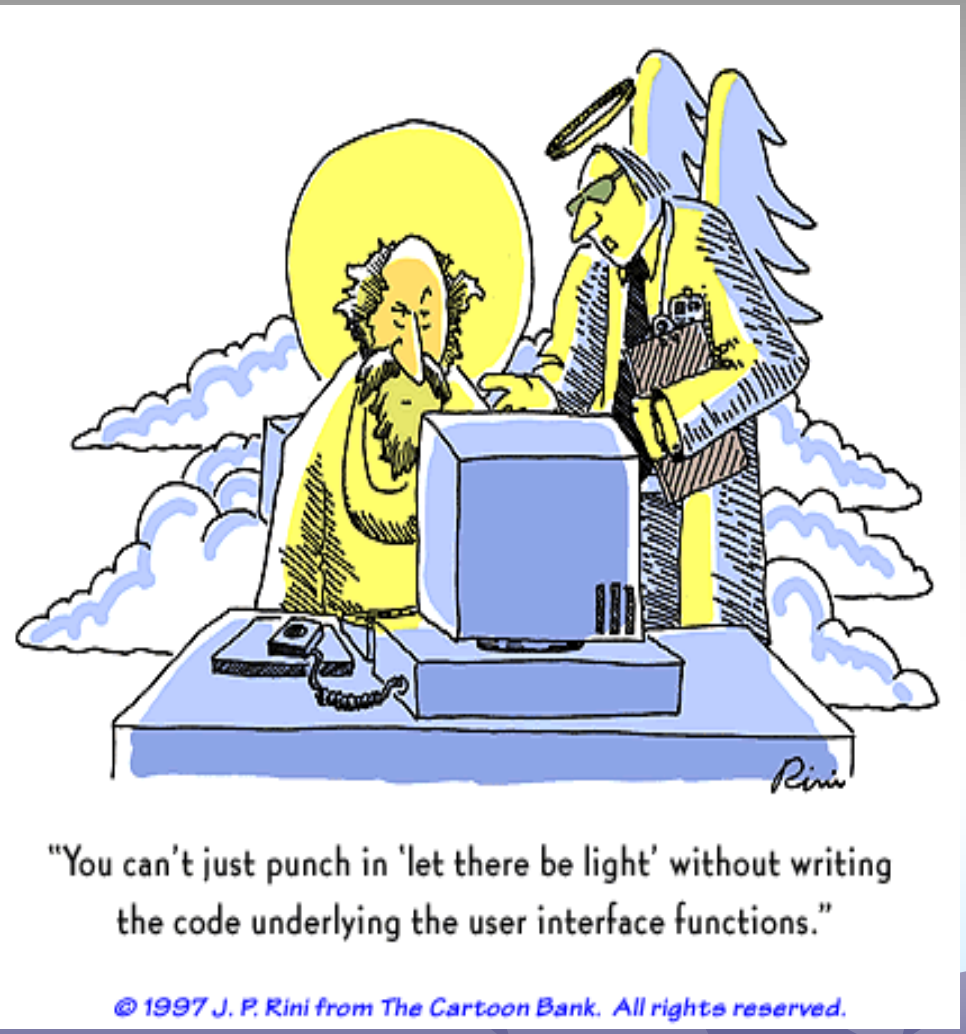
Ανθεκτικότητα:
i) *Δυνατότητα παρατήρησης*

- Κατά πόσο
- όταν ο χρήστης δίνει μια εντολή
- μπορεί να δει τα αποτελέσματα.

Ανθεκτικότητα:
ii) *Ανανηψιμότητα*



Ανθεκτικότητα:
iii) *Λειτουργικότητα*



- Οι λειτουργίες που είναι επιθυμητές από τους χρήστες
- πρέπει να μπορούν να εμφανίζονται στο σύστημα διεπαφής
- και
- να υποστηρίζονται από τον προγραμματισμό που θα υλοποιεί αυτές τις λειτουργίες.

- ***8 ΚΑΝΟΝΕΣ***
- ***ΚΑΛΟΥ ΣΧΕΔΙΑΣΜΟΥ***
- ***ΣΥΣΤΗΜΑΤΩΝ ΔΙΕΠΑΦΗΣ ΜΕ ΤΟΥΣ ΧΡΗΣΤΕΣ***

8 Κανόνες καλού σχεδιασμού:

Συνέπεια

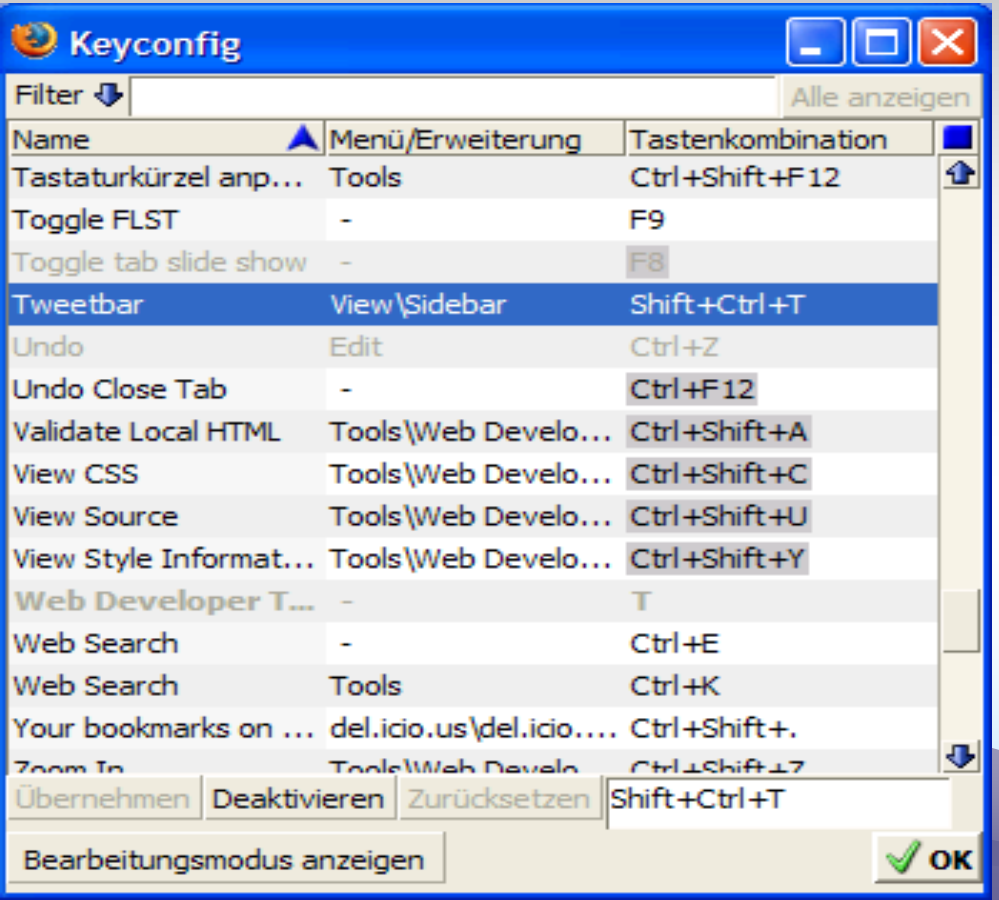
- Με βάση τις έννοιες που σχετίζονται με την ευχρηστία ενός συστήματος διεπαφής θα μπορούσαν συνοπτικά να γραφούν κάποιοι κανόνες σχεδιασμού. Οι βασικότεροι κανόνες είναι οι παρακάτω:

1) **1. Συνέπεια**

   Το σύστημα διεπαφής πρέπει να είναι συνεπές σε όλη του την έκταση (π.χ. σε χρώματα, δομή, εικονίδια, κ.λπ.)

8 Κανόνες καλού σχεδιασμού:

Shortcuts



2. **Να μπορούν οι συχνοί χρήστες να χρησιμοποιούν shortcuts**

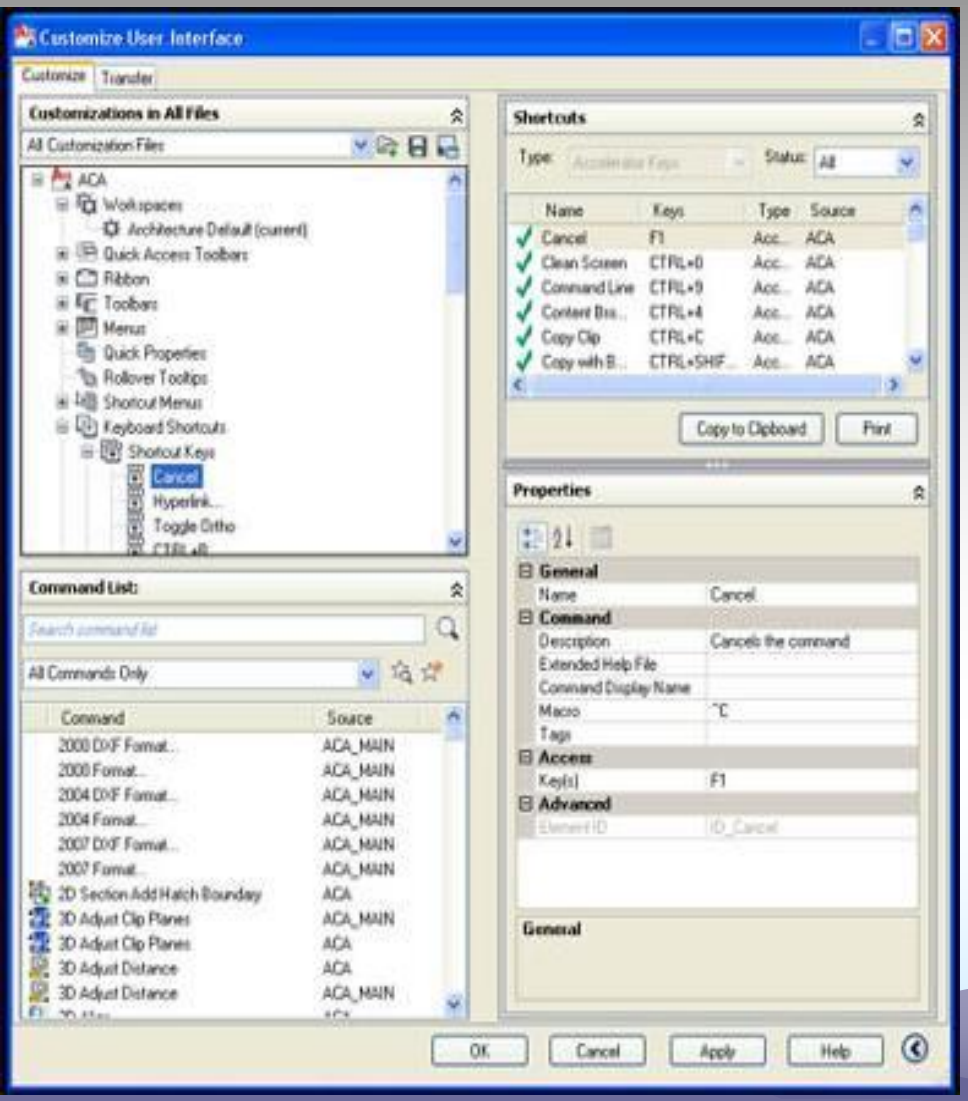
Αυτό εξασφαλίζει τη δυνατότητα να χρησιμοποιείται το πρόγραμμα από περισσότερες της μιας κατηγορίας χρηστών.

8 Κανόνες καλού σχεδιασμού:
Περιεκτικά μηνύματα

3. Ύπαρξη περιεκτικών μηνυμάτων

Τα μηνύματα προς το χρήστη πρέπει να είναι περιεκτικά και να έχουν ολόκληρη την πληροφορία που χρειάζεται ο χρήστης.

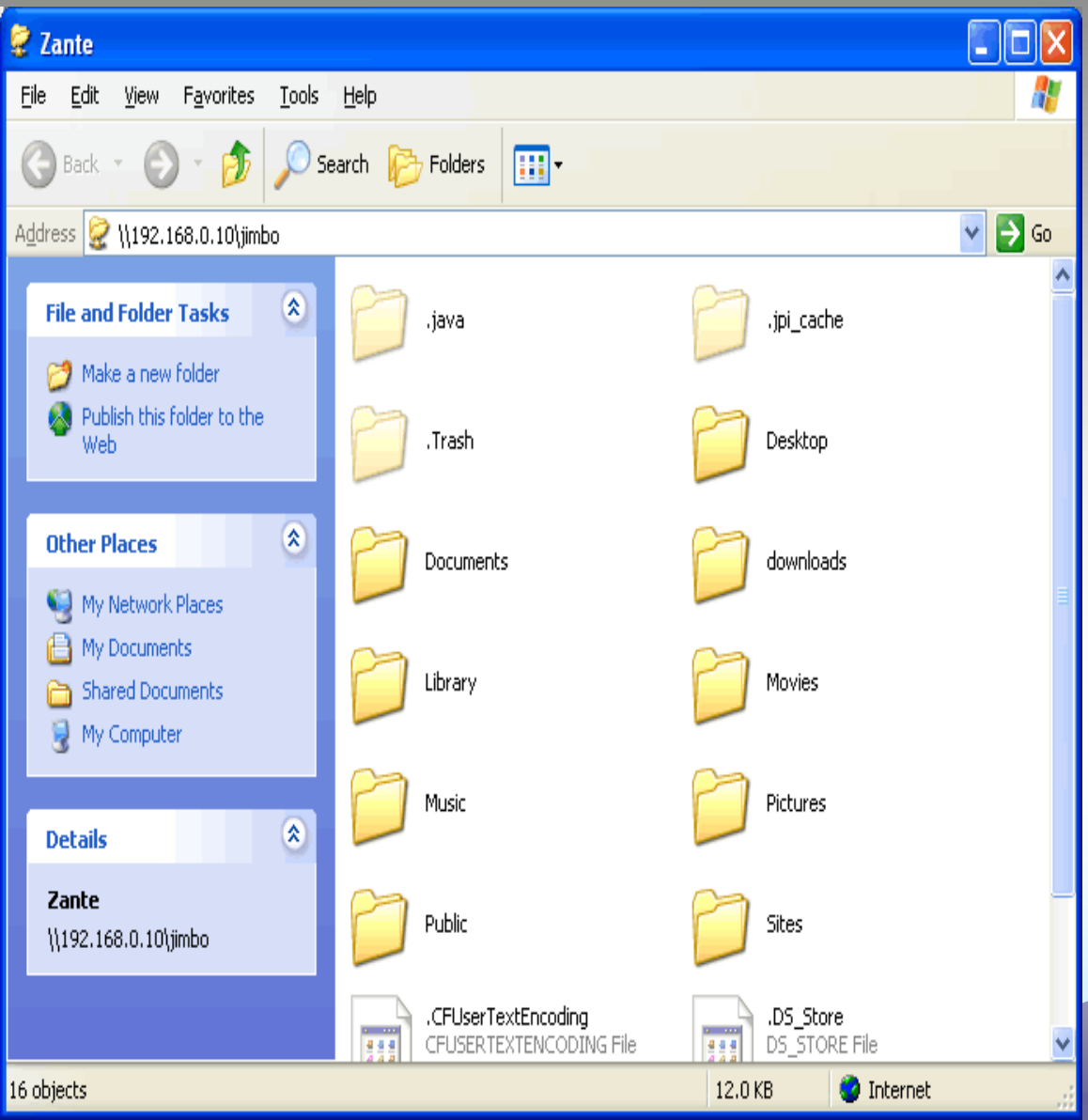
8 Κανόνες καλού σχεδιασμού:
Διάλογοι με πληρότητα



4. Σχεδιασμός διαλόγων με πληρότητα

Οι διάλογοι με το χρήστη θα πρέπει επίσης να είναι πλήρεις και περιεκτικοί.

8 Κανόνες καλού σχεδιασμού:
Αποφυγή & Εύκολη διαχείριση λαθών

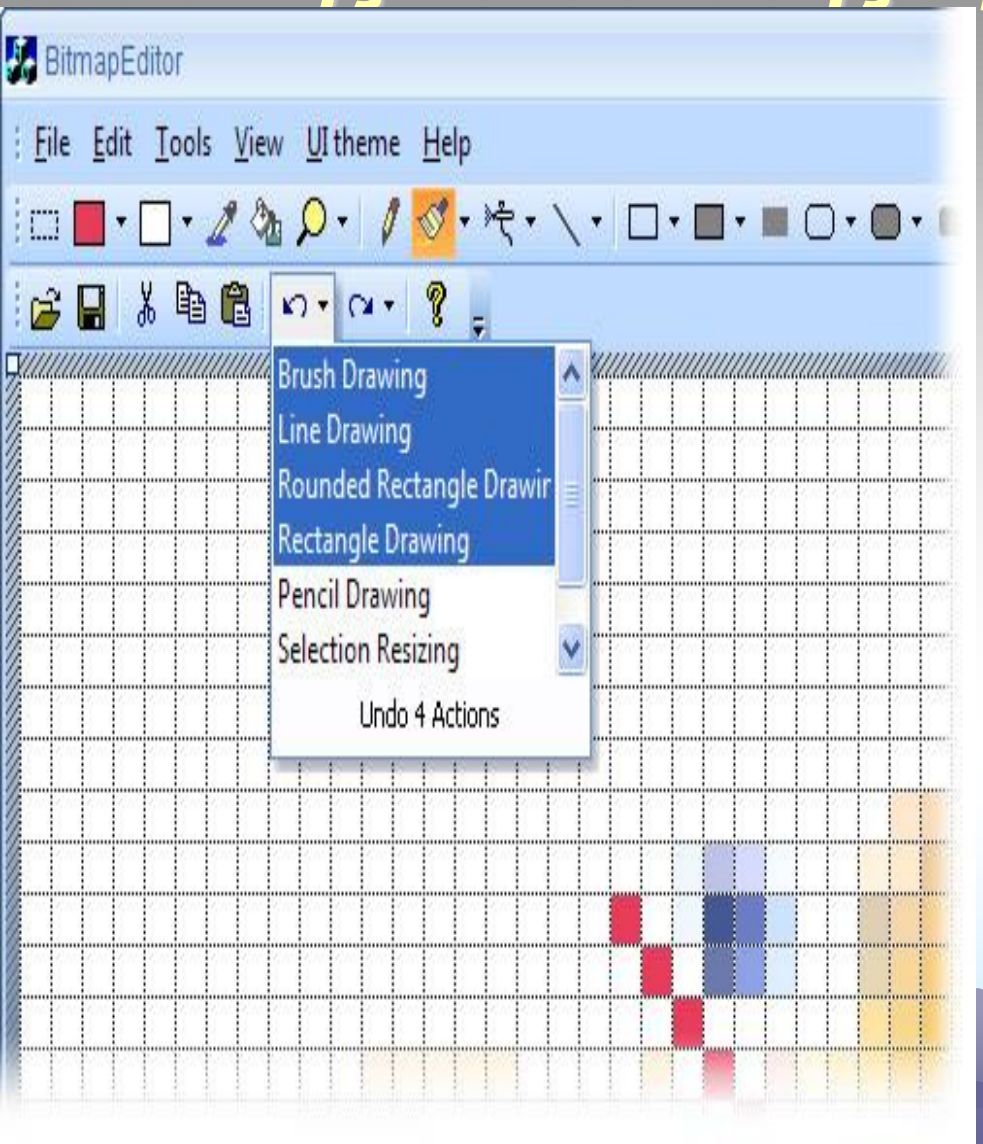


5. Δυνατότητα αποφυγής λαθών και εύκολη διαχείριση λαθών

Θα πρέπει να προβλέπονται τρόποι αποφυγής λαθών (π.χ. απενεργοποίηση επιλογών που δεν έχουν νόημα για κάποια συγκεκριμένη εργασία, όπως το κουμπί ‘back’ στην εικόνα).

Όμως στην περίπτωση που γίνει τελικά ένα λάθος από το χρήστη να μπορεί να το επανορθώσει εύκολα.

8 Κανόνες καλού σχεδιασμού:
Δυνατότητα επαναφοράς της κατάστασης πριν από ενέργεια



6. Δυνατότητα επαναφοράς της κατάστασης πριν από ενέργεια (undo)

Είναι πολύ χρήσιμο να υπάρχει η δυνατότητα επαναφοράς της κατάστασης πριν από κάποια ενέργεια. Αυτό είναι χρήσιμο γιατί ο χρήστης μπορεί να μην καταφέρει να αξιολογήσει επαρκώς την κατάσταση στην οποία βρίσκεται μετά από κάποια ενέργεια. Μπορεί επίσης να μην έχει συνειδητοποιήσει καθόλου ποιά ενέργεια τον οδήγησε στην παρούσα κατάσταση. Το αποτέλεσμα αυτών μπορεί να είναι η αδυναμία του να βρει τρόπο ανάνηψης από κάποιο λάθος που μπορεί να έκανε.

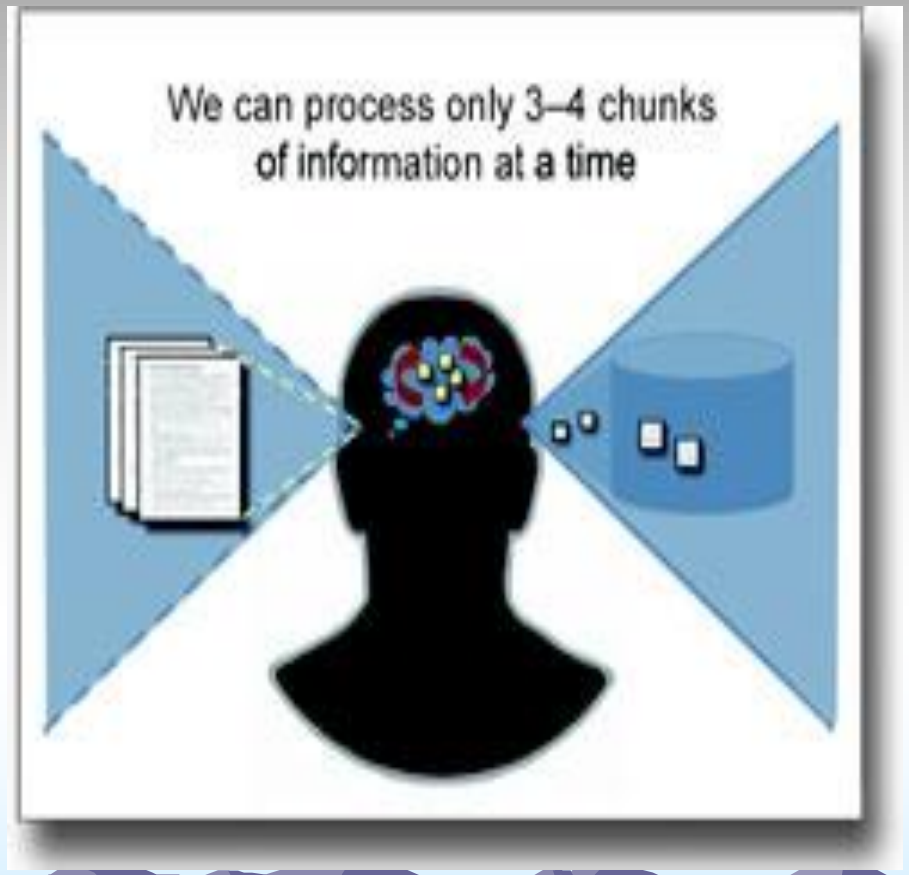
8 Κανόνες καλού σχεδιασμού:

Πρωτοβουλία του ελέγχου στο χρήστη

7. Πρωτοβουλία του ελέγχου στο χρήστη

Ο χρήστης θα πρέπει να μπορεί να έχει τον έλεγχο των ενεργειών του.

8 Κανόνες καλού σχεδιασμού:
Μείωση του φόρτου της μνήμης μικρής διάρκειας



8. Μείωση του φόρτου της μνήμης μικρής διάρκειας

Ο χρήστης δεν θα πρέπει να εξαναγκάζεται να θυμάται μια σειρά από στοιχεία για να ολοκληρώσει μία ενέργεια.