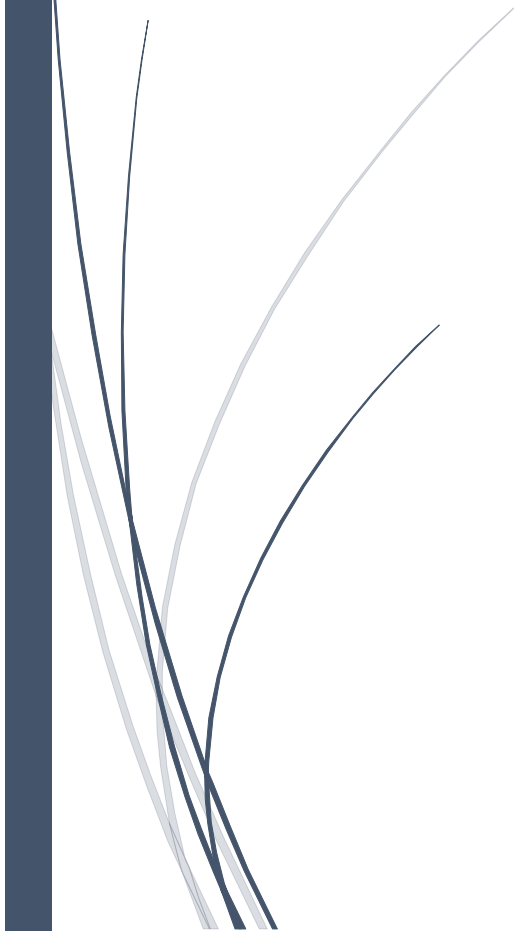
Λογισμικά CAD με Ελεύθερη Άδεια Χρήσης

ΣΜΥΡΝΗΣ ΜΑΡΙΟΣ ΑΜ:5280

ΕΠΙΒΛΕΠΩΝ: ΠΕΤΟΥΣΗΣ ΜΑΡΚΟΣ



ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας μου κύριο Μάρκο Πετούση για τη βοήθεια και τις υποδείξεις του και κυρίως για την μεγάλη υπομονή του.

Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου για την προσπάθεια και την υποστήριξη τους σε όλη την διάρκεια των σπουδών μου.

Τέλος, οικογένεια και φίλους που μέσω δικής τους παρότρυνσης και βοήθειας κατάφερα να ολοκληρώσω αυτή την εργασία.

Περίληψη

Το λογισμικό CAD είναι ένα από πιο δυνατά εργαλεία του μηχανικού και έχει αλλάξει σημαντικά τον τρόπο ανάπτυξης των προϊόντων τις τελευταίες δεκαετίες. Τα διαθέσιμα εμπορικά λογισμικά έχουν πλέον πολύ μεγάλες δυνατότητες και εργαλεία για να υποστηρίξουν όλα τα βήματα του κύκλου ανάπτυξης και του κύκλου ζωής του προϊόντος. Τα τελευταία χρόνια έχουν παρουσιαστεί στο διαδίκτυο λογισμικά CAD με ελεύθερη άδεια χρήσης, τα οποία παρέχουν ένα μεγάλο μέρος από τα σχεδιαστικά εργαλεία που υπάρχουν στα εμπορικά λογισμικά για σχεδίαση σε δύο και τρεις διαστάσεις, χωρίς κόστος. Κάποια από αυτά είναι λογισμικά ανοικτού κώδικα και άλλα είναι λογισμικά που παρέχονται δωρεάν από εταιρίες λογισμικού CAD, για να υποστηρίξουν την εμπορική τους πολιτική.

Αντικείμενο της πτυχιακής εργασίας είναι η διερεύνηση όλων των διαθέσιμων λογισμικών CAD με ελεύθερη άδεια χρήσης. Στα πλαίσια της εργασίας θα αναζητηθούν και θα αναλυθούν όλα τα διαθέσιμα λογισμικά CAD ελεύθερης χρήσης ως προς τις δυνατότητες τους, την ευκολία χρήσης τους, την αξιοπιστία τους και γενικά όλες τις παραμέτρους αξιολόγησης ενός CAD λογισμικού.

Αρχικά, θα διερευνηθεί η απαιτούμενη διαδικασία για την αξιολόγηση ενός λογισμικού CAD. Θα αναζητηθούν οι διαθέσιμες διαδικασίες για αυτό το σκοπό στο διαδίκτυο και θα επιλεγεί αυτή που θα κριθεί ως πιο κατάλληλη για το αντικείμενο της παρούσης εργασίας. Αυτή θα προσαρμοστεί κατάλληλα και στη συνέχεια θα

εφαρμοστεί στα λογισμικά CAD με ελεύθερη άδεια χρήσης που είναι διαθέσιμα στο διαδίκτυο. Με βάση αυτή την ανάλυση θα προκύψουν χρήσιμα συμπεράσματα σε σχέση με τις δυνατότητες των λογισμικών αυτών και το πιο λογισμικό είναι καλύτερο.

Τα καλύτερα λογισμικά για σχεδίαση σε δύο και σε τρεις διαστάσεις θα συγκριθούν με αντίστοιχα εμπορικά λογισμικά, ακολουθώντας την ίδια μεθοδολογία. Εάν κάτι τέτοιο δεν είναι εφικτό, θα αναπτυχθούν δύο πρότυπα σχέδια, ένα σε δύο και ένα σε τρεις διαστάσεις. Αυτά θα μοντελοποιηθούν σε αντίστοιχα εμπορικά λογισμικά CAD και στα αντίστοιχα λογισμικά με άδεια ελεύθερης χρήσης, ώστε να προκύψουν χρήσιμα συμπεράσματα σε σχέση με τις δυνατότητες αυτών των λογισμικών, συγκρινόμενα με αντίστοιχα εμπορικά.

Abstract

CAD software is one of the most powerful engineering tools and has significantly changed the way products are developed over the last decades. Available commercial software now has great potential and tools to support all steps of the product development cycle and life cycle. In the last few years, free CAD software has been presented on the internet, providing a great deal of design tools in commercial software for design in two and three dimensions at no cost. Some of these are open source software and others are software provided free of charge by CAD software companies to support their commercial policy.

The aim of the dissertation is to investigate all available CAD software with free license. All available free CAD software will be searched for and analyzed in terms of their capabilities, ease of use, reliability and generally all CAD software evaluation parameters.

Initially, the required procedure for evaluating a CAD software will be explored. The available procedures for this purpose will be searched on the Internet and the one selected as the most appropriate for the subject of this paper will be selected. This will be customized and then applied to free CAD CAD software available on the Internet. Based on this analysis, useful conclusions will be drawn with regard to the capabilities of these softwares and most software is better.

The best software for design in two and three dimensions will be compared to corresponding commercial software, following the same methodology. If this is not feasible, two standard drawings, one in two and one in three dimensions, will be developed. These will be modeled on corresponding commercial CAD software and the corresponding software with a free-to-use license to provide useful conclusions with regard to the capabilities of these softwares compared to commercially available ones.

1 Περιεχόμενα

1 Περιεχόμενα 6

1.1. Εισαγωγή 7

2. Βιβλιογραφική Ανασκόπηση 10

2.1. Ορισμός Ελεύθερου Λογισμικού 10

2.2. Ορισμός Λογισμικού Ανοικτού Κώδικα 13

2.3. Ιστορία του Ελεύθερου Λογισμικού και του Λογισμικού Ανοικτού Κώδικα 16

2.4. Το GNU project 23

2.5. Το FSF 27

2.6. Εταιρία Ελεύθερου Λογισμικού και οι Στόχοι 29

2.7. Πλεονεκτήματα Ελεύθερου Λογισμικού 29

2.8. Μειονεκτήματα Ελεύθερου Λογισμικού 32

2.9. Διαφορές Λογισμικού Ανοικτού και Κλειστού Κώδικα 33

2.10. Άδειες Χρήσης Λογισμικού 36

2.11. Άδειες Χρήσης Ελεύθερου Λογισμικού 40

2.12. Τι είναι εκπαιδευτικό Λογισμικό 44

2.13. Κατηγορίες Εκπαιδευτικού Λογισμικού 45

2.14. Λογισμικά CAD 48

2.14.1. Ορισμός 48

2.14.2. Χρήσεις 52

2.14.3. Τύποι 55

2.14.4. Σύγκριση και αξιολόγηση λογισμικών 85

2.14.5 Εφαρμογές 87

3. Συμπεράσματα 92

4. Βιβλιογραφία 95

1.1. Εισαγωγή

Στην πλειονότητα των περιπτώσεων, τα open source λογισμικά είναι δωρεάν, όπως το Android, το Linux και το Mozilla Firefox, κλπ. Σήμερα, το λογισμικό ανοιχτού κώδικα έχει γίνει ένα αναπόσπαστο και σημαντικό μέρος των υπολογιστών και της σημερινής τεχνολογίας. Πολλά επιτυχημένα προγράμματα λογισμικού ανοιχτού κώδικα όπως το Linux, το Apache, το Eclipse, το OpenCV, κ.λ.π, εξαιτίας της άριστης ποιότητας και της προσιτότητας στο κοινό, αντιμετωπίζονται ως de facto βιομηχανικό λογισμικό πρότυπο, που χρησιμοποιείται ευρέως σε διάφορους τομείς, με αποτέλεσμα να παρέχει τεράστια κοινωνική αξία και συνεισφορά στην επιχειρηματικότητα.



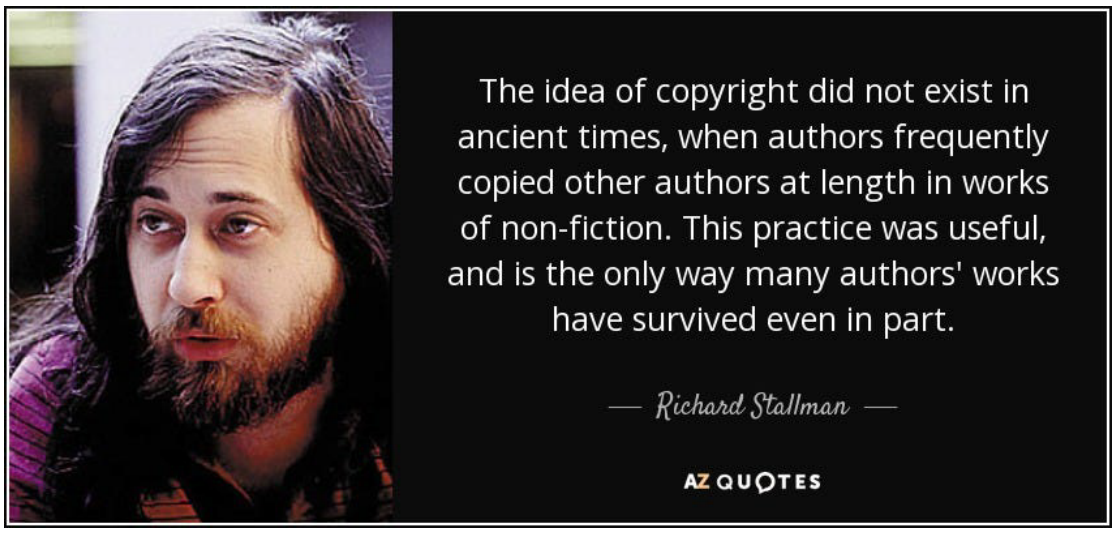
1.1. Τα λογότυπα των ελευθέρας χρήσης λογισμικών Apache, Linux και Eclipse, αντίστοιχα

Η ιστορία της ανάπτυξης λογισμικού είναι ένας μικρόκοσμος της ιστορίας. Από την αρχική περίοδο, ήταν μια πρωτόγνωρη ελευθερία για τη μικρή μειονότητα των προγραμματιστών, που είχαν την ελευθερία να τροποποιούν και να μοιράζονται το διαθέσιμο λογισμικό. Στη συνέχεια υπήρξε μια ισχυρή εποχή που μπορεί να θεωρηθεί ως σπουδαία δουλειά του Μπιλ Γκέιτς, επειδή οδηγούσε την παλίρροια του εμπορικού λογισμικού.



1.2. Οι ιδρυτές της Microsoft Paul Allen και Bill Gates, στις αρχές της δεκαετίας του '80 αποφάσισαν να δημιουργήσουν λογισμικό, αντί για υπολογιστή

Ο Richard Stallman, ο ηγέτης του ελεύθερου λογισμικού, ξεκίνησε αργότερα τον "ρομαντικό διαφωτισμό". Αργότερα κυκλοφόρησε ένα κύμα ανοιχτού κώδικα, το οποίο έφερε μια καλύτερη ισορροπία μεταξύ εμπορίου και ελευθερίας. Πρωτοπόροι στη βιομηχανία λογισμικού, καθώς και οι ηγέτες της ανθρώπινης κοινωνίας, διερευνούν το ιδανικό και την πραγματικότητα της τέλειας σύντηξης δρόμου και στροφών.



1-1.3. Richard Stallman, δημιουργός του GNU και ιδρυτής του Ιδρύματος Ελεύθερου Λογισμικού (FSF)

Το λογισμικό ανοικτής πηγής αναπτύσσεται κυρίως από ομάδες προγραμματιστών διάσπαρτες σε όλο τον κόσμο, αλλά συγχρόνως συμμετέχουν επίσης και πανεπιστήμια, κυβερνητικές υπηρεσίες, εργολάβοι, εταιρείες και εμπορικές εταιρείες. Με τον ανταγωνισμό και τις πιέσεις στο τρέχον περιβάλλον λογισμικού, όλο και περισσότερες εταιρείες θα ήθελαν το λογισμικό ανοιχτού κώδικα, λόγω των ποικίλων πλεονεκτημάτων του, όπως η μείωση του κόστους, η βελτίωση της ανταγωνιστικής δύναμης και της αποτελεσματικότητας κ.ο.κ. Με βάση τα παραπάνω, η συγκεκριμένη εργασία θα αναλύσει το περιεχόμενο των λογισμικών CAD με ελεύθερη άδεια χρήσης.

2. Βιβλιογραφική Ανασκόπηση

2.1. Ορισμός Ελεύθερου Λογισμικού

Μεταξύ των κυριότερων τύπων λογισμικού (εμπορικά, shareware, adware κ.λπ.), το λογισμικό freeware και το ανοικτού κώδικα, είναι μοναδικά. Το δωρεάν λογισμικό είναι λογισμικό που προστατεύεται από πνευματικά δικαιώματα και παρέχεται από τον κάτοχό του για χρήση από άλλους. Ο συντάκτης, διατηρεί την αποκλειστική κατοχή των πνευματικών δικαιωμάτων, οπότε οι χρήστες δεν μπορούν να αλλάξουν το λογισμικό. Οι δημιουργοί δωρεάν λογισμικού επιτρέπουν σε άτομα να χρησιμοποιούν τις παραγωγές τους με οποιοδήποτε νόμιμο τρόπο, αλλά δεν επιτρέπουν σε κανέναν να πουλήσει το λογισμικό για κέρδος. Επιπλέον, πολλοί προμηθευτές δωρεάν λογισμικού επιβάλλουν όρια στη χρήση των προϊόντων τους, περιορίζοντας την εφαρμογή τους στις εμπορικές προσπάθειες, για παράδειγμα. Το λογισμικό ανοιχτού κώδικα μεταφέρει την έννοια του δωρεάν λογισμικού στο τελικό συμπέρασμα. Με τα προϊόντα ανοιχτού κώδικα, ο κάτοχος των πνευματικών δικαιωμάτων δίνει στους άλλους το δικαίωμα να μελετούν, να τροποποιούν και να διανέμουν το λογισμικό δωρεάν σε οποιονδήποτε για οποιονδήποτε σκοπό. Πολύ συχνά, τα προϊόντα ανοιχτού κώδικα προέρχονται από τις συνεργατικές προσπάθειες συνεργατών που ζουν σε πολυάριθμες τοποθεσίες σε όλο τον κόσμο. Ο κώδικας ακατέργαστου προγράμματος μαζί με το πρόγραμμα που έχει καταρτιστεί είναι διαθέσιμος σε οποιονδήποτε είναι πρόθυμος να το αποκτήσει, να το ελέγξει και να κάνει προσθήκες, ή βελτιώσεις σε αυτό, αναμένοντας ότι οι συνδυασμένες προσπάθειες πολλών ανθρώπων θα οδηγήσουν σε ένα προϊόν όλο και πιο χρήσιμο και αξιόπιστο στους τελικούς χρήστες (Alexander, Christopher. 1965).

Αν και ορισμένα προϊόντα ανοιχτού κώδικα στερούνται τεκμηρίωσης, πολλά (αν όχι τα περισσότερα) διαθέτουν ενεργές ομάδες χρηστών ή κοινότητες που χρησιμεύουν ως πηγές βοήθειας προς τους χρήστες. Για να θεωρηθεί ως ανοικτή πηγή, ένα προϊόν λογισμικού πρέπει να πληρεί διάφορα κριτήρια, μεταξύ των οποίων είναι τα εξής (Amidon, Jane. 2012):

- Το λογισμικό πρέπει να είναι ελεύθερα διαθέσιμο χωρίς κόστος, δικαιώματα ή τέλη οποιουδήποτε είδους.
- Το πρόγραμμα πρέπει να διανέμεται ως πηγαίος κώδικας (για τους προγραμματιστές) και ως κατερτισμένος κώδικας (για τους τελικούς χρήστες).
- Οι τελικοί χρήστες και οι προγραμματιστές ενδέχεται να αλλάξουν τον κώδικα του προγράμματος.
- Ο τροποποιημένος πηγαίος κώδικας πρέπει να αναδιανεμηθεί υπό τους ίδιους όρους με την άδεια για την αρχική έκδοση του λογισμικού.

Οι χρήστες πρέπει να προστατεύονται από τον πειρασμό να εξομοιώνουν το δωρεάν λογισμικό και το λογισμικό ανοιχτού κώδικα με το λογισμικό public domain. Αυτή η τελευταία κατηγορία λογισμικού δεν προστατεύεται από πνευματικά δικαιώματα και ως εκ τούτου, είναι απαλλαγμένο από κάθε κόστος και μπορεί να χρησιμοποιηθεί χωρίς κανένα περιορισμό. Το Freeware και το λογισμικό ανοιχτού κώδικα διαθέτει πνευματικά δικαιώματα. Παρόλο που πολλοί χρήστες μπορεί να μην είναι εξοικειωμένοι με τις έννοιες του δωρεάν λογισμικού και του λογισμικού ανοιχτού κώδικα, εξακολουθούν να βασίζονται καθημερινά στις συνήθεις εργασίες τους. Εάν κάποιος χρησιμοποιεί τον Παγκόσμιο Ιστό, πρέπει να χρησιμοποιεί ένα πρόγραμμα περιήγησης στο Internet. Όσοι χρησιμοποιούσανε τον Internet Explorer της Microsoft

χρησιμοποιούσανε ένα από τα πιο δημοφιλή προϊόντα δωρεάν λογισμικού, ενώ οι χρήστες του Firefox βασίζονται σε ένα πακέτο ανοιχτού κώδικα (Babar, A. 2016).





2.1.1. Διάσημοι περιηγητές του διαδυκτίου, ο Firefox της Mozilla κι ο Internet Explorer της Microsoft

Πολλά συστήματα freeware και ανοιχτού κώδικα είναι πρότυπα στους τομείς τους. Το CCleaner (δωρεάν λογισμικό) έχει κερδίσει πολλά βραβεία για την αποτελεσματικότητά του όχι μόνο ως εργαλείο καθαρισμού και βελτιστοποίησης συστημάτων, αλλά και ως θεματοφύλακα της ιδιωτικής ζωής των χρηστών. Το Audacity (ανοιχτού κώδικα) είναι ηχογράφηση και συντάκτης που χρησιμοποιείται όχι μόνο από ερασιτέχνες αλλά και από επαγγελματίες στο πεδίο της μουσικής. Η πρόσβαση στον Παγκόσμιο Ιστό θα ήταν αδύνατη χωρίς τη χρήση του Apache (ανοιχτού κώδικα), του διακομιστή HTTP. Το δωρεάν λογισμικό μπορεί να γίνει shareware, το εμπορικό λογισμικό μπορεί να μετατραπεί σε δωρεάν λογισμικό, κλπ (Coppinger, J. 2017).





2.1.2. Δωρεάν διαθέσιμα λογισμικά. Το CC Cleaner λογισμικό καθαρισμού ιών/malware και το Audacity λογισμικό ηχογράφησης ανοιχτού κώδικα

2.2. Ορισμός Λογισμικού Ανοικτού Κώδικα

Στη δεκαετία του 1970, το ιδιόκτητο λογισμικό - δηλαδή το λογισμικό που δεν επιτρέπει στους χρήστες να το αναδιανέμουν, να το τροποποιήσουν, ή να αποκτήσουν πρόσβαση στον πηγαίο κώδικα του - έγινε μια κοινά αποδεκτή κατάσταση. Η ανάπτυξη λογισμικού ανοιχτού κώδικα ήταν μια αντίδραση στο γεγονός ότι δεν μπορούν να γίνουν αλλαγές, ή βελτιώσεις σε ιδιόκτητο λογισμικό από άλλους προγραμματιστές, ή χρήστες. Το κίνημα ανοιχτής πηγής ξεκίνησε με το μοντέλο γενικής δημόσιας άδειας του Richard Stallman (στη δεκαετία του '80), το οποίο δήλωνε ότι το λογισμικό πρέπει να τροποποιείται ελεύθερα, με την προϋπόθεση ότι εάν γίνουν βελτιώσεις στο λογισμικό, πρέπει να επαναφερθούν οι βελτιώσεις στην ανοιχτή πηγή. Το σκεπτικό για την κίνηση ανοιχτού κώδικα είναι ότι μια μεγαλύτερη ομάδα προγραμματιστών που δεν ασχολείται με την κυριότητα ιδιοκτησίας, θα παράγει ένα καλύτερο προϊόν (Babar, A. 2016).



2.2.1. Το λογότυπο του Κινήματος Ελεύθερου Κώδικα

Το λογισμικό ανοιχτού κώδικα είναι ένα λογισμικό υπολογιστή που διαθέτει πηγαίο κώδικα για το ευρύ κοινό για χρήση ως έχει, ή με τροποποιήσεις. Αυτό το λογισμικό συνήθως δεν απαιτεί τέλη αδείας. Υπάρχουν εφαρμογές λογισμικού ανοιχτού κώδικα για διάφορες χρήσεις όπως αυτοματισμοί γραφείου, σχεδιασμός ιστοσελίδων, διαχείριση περιεχομένου, λειτουργικά συστήματα και επικοινωνίες. Το βασικό γεγονός που κάνει το λογισμικό ανοιχτού κώδικα (OSS) διαφορετικό από το ιδιόκτητο λογισμικό, είναι η άδειά του. Ως υλικό πνευματικών δικαιωμάτων το λογισμικό είναι σχεδόν πάντα αδειοδοτημένο. Η άδεια χρήσης υποδεικνύει πώς μπορεί να χρησιμοποιηθεί το λογισμικό. Το OSS είναι μοναδικό, δεδομένου ότι κυκλοφορεί πάντοτε με άδεια που έχει πιστοποιηθεί ότι πληρεί τα κριτήρια του ορισμού ανοικτού κώδικα. Αυτά τα κριτήρια περιλαμβάνουν το δικαίωμα (3D Innovations. 2013):

- Αναδιανομή του λογισμικού χωρίς περιορισμούς.
- Απόκτηση πρόσβασης στον πηγαίο κώδικα.

- Τροποποίηση του πηγαίου κώδικα.
- Διανομή της τροποποιημένης έκδοσης του λογισμικού.

Αντίθετα, οι δημιουργοί ιδιόκτητου λογισμικού συνήθως δεν κάνουν τον πηγαίο κώδικα τους διαθέσιμο σε άλλους για να τροποποιήσουν. Όταν εξετάζονται τα πλεονεκτήματα του λογισμικού ανοιχτού κώδικα, θα πρέπει να εξεταστεί το ίδιο το προϊόν ανοικτού κώδικα.

Τα προϊόντα ανοικτής πηγής ποικίλλουν ως προς την ποιότητα τους. Το λογισμικό OSS δεν συνοδεύεται από τηλεφωνική υποστήριξη, ή εξατομικευμένη υποστήριξη μέσω ηλεκτρονικού ταχυδρομείου. Ωστόσο, υπάρχουν πάροχοι εμπορικών υπηρεσιών που παρέχουν υποστήριξη. Εάν υπάρχει μεγάλη υποστήριξη, αξίζει να εξεταστεί εάν το συνολικό κόστος χρήσης ενός προϊόντος ανοικτού πηγαίου κώδικα θα είναι υψηλότερο από αυτό ενός ιδιόκτητου προϊόντος (Coppinger, J. 2017).

Το λογισμικό ανοιχτού κώδικα είναι μοναδικό, δεδομένου ότι κυκλοφορεί πάντοτε με άδεια χρήσης που επιτρέπει στους χρήστες να έχουν πρόσβαση, να τροποποιούν και να αναδιανέμουν τον πηγαίο κώδικα. Ο πηγαίος κώδικας είναι μια εξειδικευμένη γλώσσα που επιτρέπει στους προγραμματιστές λογισμικού να δημιουργούν και να τροποποιούν προγράμματα υπολογιστών. Αν δεν υπάρχει νόμιμη πρόσβαση στον πηγαίο κώδικα, τότε το πρόγραμμα δεν μπορεί να αλλάξει ή να μεταφερθεί σε διαφορετικό υπολογιστή (Amidon, Jane. 2012).

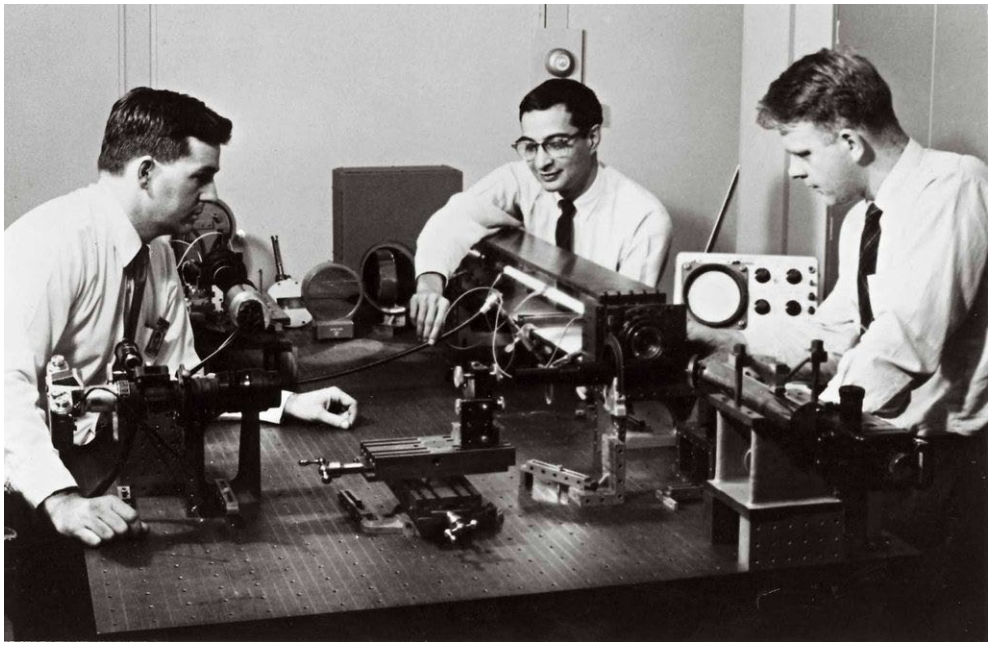
2.3. Ιστορία του Ελεύθερου Λογισμικού και του Λογισμικού Ανοικτού Κώδικα

Το ανοικτό λογισμικό δεν θα ήταν τίποτα χωρίς τον κοινόχρηστο πηγαίο κώδικα. Ο πηγαίος κώδικας είναι απαραίτητος για το άνοιγμα του λογισμικού λόγω των ιστορικά συγκεκριμένων τρόπων με τους οποίους έχει γίνει κοινή χρήση. Για την ανταλλαγή κωδικοποιητικών υλικών και την βαθύτερη κατανόηση του λειτουργικού συστήματος της IBM, η ομάδα μελέτης με την επωνυμία IBM SHARE GROUP SHARE δημιουργήθηκε το 1955.



2.3.1. Το 1955 το πρώτο δίκτυο υπολογιστών ονόματι Share δημιουργήθηκε από χρήστες του μοντέλου υπολογιστή 704 της IBM

Η ιστορία των κανόνων για την κοινή χρήση πηγαίου κώδικα δεν είναι τυχαία και η ιστορία του λειτουργικού συστήματος UNIX. Το σύστημα UNIX που αναπτύχθηκε από την Bell Labs, ήταν δωρεάν για πανεπιστήμια και επιστημονικούς φορείς. Αυτά τα ιδρύματα και το ερευνητικό κέντρο βελτίωσαν το εν λόγω σύστημα και άνοιξαν τους πηγαίους κώδικες του συστήματος. Έπειτα από αυτό, λόγω της χρήσης του UseNet, προέκυψε η κατάσταση της κοινής χρήσης κώδικα.



2.3.2. Bell Labs Περιβάλλον καινοτομιών και το θαύμα του μέλλοντος

Έγινε σύνδεση με το συνδεδεμένο σύστημα UNIX και την κοινότητα προγραμματισμού. Η κοινή χρήση του πηγαίου κώδικα βελτίωσε την εξέλιξη της τεχνολογίας σε νέα ύψη. Υπάρχουν τρεις κύριες δραστηριότητες σχετικά με τις συνεργατικές εξελίξεις (Babar, A. 2016):

- Το Ίδρυμα Ελεύθερου Λογισμικού (FSF) είναι ένας μη κερδοσκοπικός οργανισμός που ιδρύθηκε από τον Richard Stallman στις 4 Οκτωβρίου 1985 για να υποστηρίξει το κίνημα του ελεύθερου λογισμικού.
- Το έργο GNU ξεκίνησε το 1984 για να αναπτύξει το σύστημα GNU. Το GNU είναι σύστημα που μοιάζει με Unix και είναι σύντομο για το "GNU's Not Unix!". Πρόκειται για ένα ελεύθερο λογισμικό και μπορεί να ανταποκριθεί στη χρήση γαλακτοκομικών.
- Για να χρησιμοποιήσουν το λογισμικό GNU, οι προγραμματιστές έπρεπε να συμφωνήσουν για την άδεια του να κωδικοποιούν με την GPL.

- Αυτές οι συνεργατικές εξελίξεις, είτε αναφορές, σηματοδοτούν την αρχή της ιδέας διαμόρφωσης του ανοιχτού λογισμικού.



2.3.3. Λογότυπο των Unix

Σε αυτή τη φάση η ανοικτή πηγή άρχισε να γίνεται διάσημη, αυτό οφείλεται κυρίως στο Linux. Το σύστημα Linux αναπτύχθηκε από τον Linus Torvalds ο οποίος σπούδασε στο Ελσίνκι το 1991 και το ίδιο έτος η έκδοση 0.02 Linux κυκλοφόρησε με άδεια αδειοδοτημένου GPL. Μετά από τόσους πολλούς προγραμματιστές και χρήστες, βελτιώθηκε, το Linux 1.0 κυκλοφόρησε το 1994.

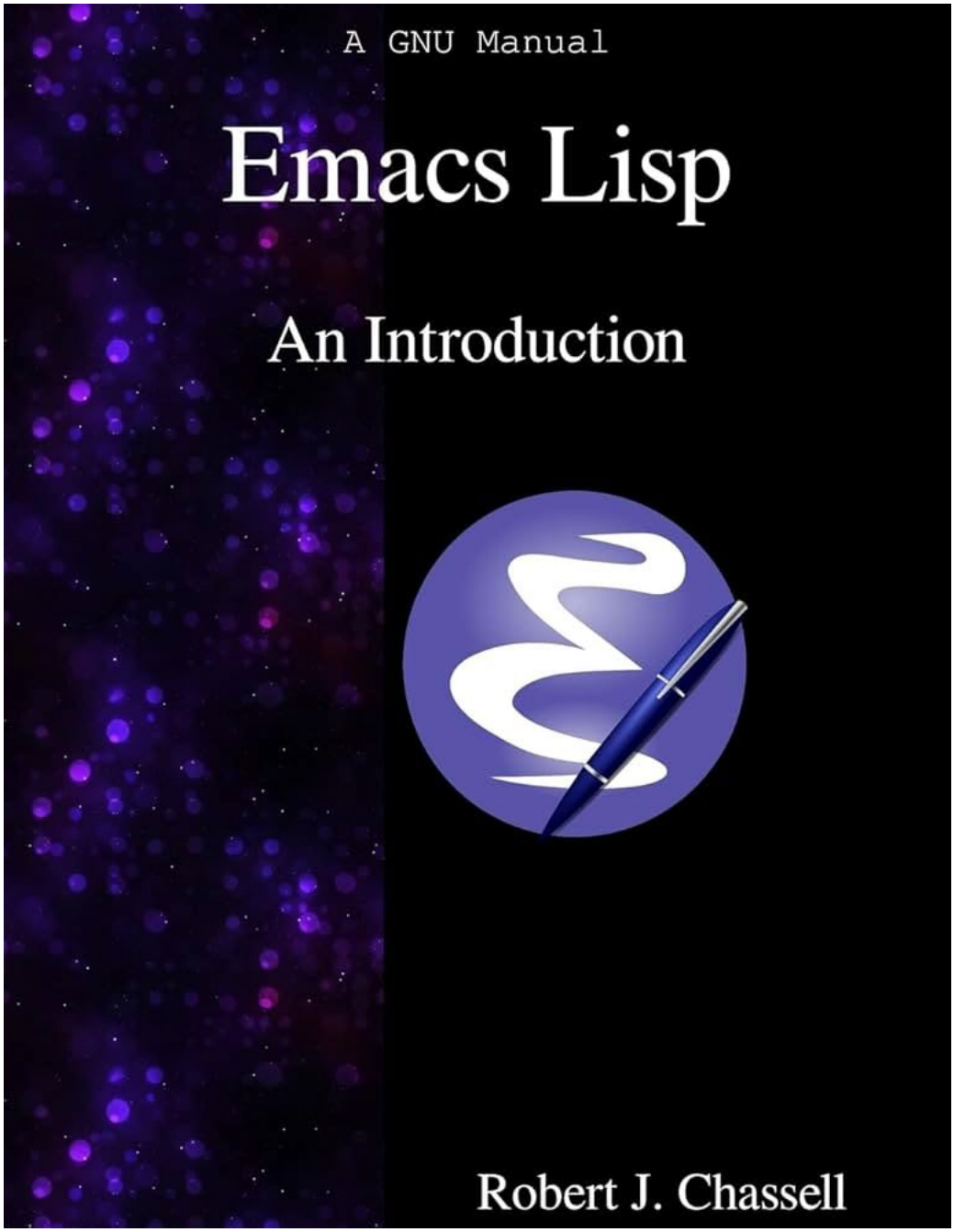


2.3.4. 1991, Ο Linus Torvalds πρωτοκυκλοφορεί τα Linux

Ενάντια στην πτώση της ανακοίνωσης ενός πηγαίου κώδικα για το Navigator που εκδόθηκε από τον Netscape τον Ιανουάριο του 1998, μερικοί άνθρωποι στο κίνημα του ελεύθερου λογισμικού σε μια συνεδρίαση στρατηγικής στην Καλιφόρνια εισήγαγαν για πρώτη φορά την έννοια του "open source". Όσον αφορά το ελεύθερο λογισμικό, στην πραγματικότητα, υπάρχει μια άλλη κίνηση που είναι επίσης γνωστή, η κίνηση λογισμικού ανοιχτού κώδικα. Είναι ένας κλάδος της κίνησης του ελεύθερου λογισμικού, αλλά η διαφορά δεν είναι προφανής (Coppinger, J. 2017).

Σύμφωνα με την έρευνα του Godfrey και του Tu (2000), έδειξαν ότι πληρούν τις δημοσιευμένες μελέτες, ή έρευνες που σχετίζονται με την εξέλιξη του λογισμικού που εκτελείται, ή πραγματοποιείται σε μια ενιαία εταιρεία που αξιοποίησε τις παραδοσιακές τεχνικές ανάπτυξης και διαχείρισης της γνώσης (Godfrey & Tu, 2000). Ως εκ τούτου, με τον ανταγωνισμό στο τρέχον περιβάλλον λογισμικού αναπτύσσεται, ή έχει αναπτυχθεί μια μεγάλη ποικιλία λογισμικού ανοιχτού κώδικα. Τα Linux είναι το πιο διάσημο λειτουργικό σύστημα στην ανάπτυξη λογισμικού ανοιχτού κώδικα, αλλά είναι επίσης ένα τυπικό παράδειγμα από τα πιο επιτυχημένα λειτουργικά συστήματα ανοιχτού κώδικα. Από την άλλη, ένα από τα χαρακτηριστικά των γλωσσών προγραμματισμού είναι η ανοιχτή πηγή. Επιπλέον οι γλώσσες προγραμματισμού είναι το θεμέλιο της ανάπτυξης λογισμικού, ακόμη και η ανάπτυξη λογισμικού ανοιχτού κώδικα. Οι περισσότεροι χρήστες του λογισμικού ανοικτής πηγής είναι οι προγραμματιστές που εφαρμόζουν τη γλώσσα προγραμματισμού κάθε φορά (Babar, A. 2016).

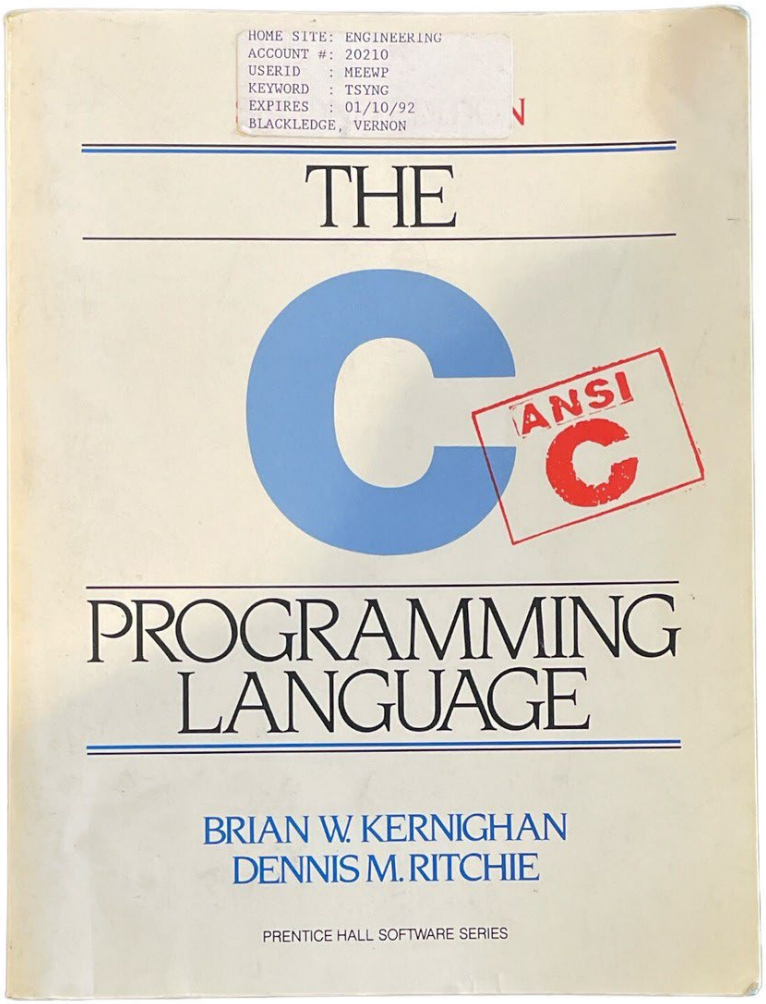
Ο χρόνος είναι σαν ένα λεωφορείο, η ανάπτυξη του δικτύου υπολογιστών έχει ξεκινίσει να αναπτύσεται εδώ και είκοσι χρόνια. Η γλώσσα προγραμματισμού είναι από τα εργαλεία που οι προγραμματιστές μπορούν να ελέγξουν εύκολα στον υπολογιστή, στους διακομιστές, στο δίκτυο και στα προγράμματα. Γενικά, η γλώσσα προγραμματισμού είναι ανοικτή ως επί το πλείστον. Αυτό σημαίνει ότι όλοι οι προγραμματιστές, ή ακόμη και οι χρήστες μπορούν να προσθέσουν κάθε γλώσσα προγραμματισμού στα προγράμματα, ή τους διακομιστές τους. Ωστόσο, λόγω της εξέλιξης του υπολογιστή και του δικτύου, η γλώσσα προγραμματισμού έκανε επίσης την εξέλιξη και την καινοτομία. Για παράδειγμα, το Emacs Lisp που ήταν η πιο δημοφιλής ανοικτή γλώσσα προγραμματισμού το 1993. Το Emacs Lisp προέρχεται από τις παλαιές γλώσσες προγραμματισμού Lisp. Λόγω της δυνατότητας επέκτασης του Emacs, οι περισσότεροι προγραμματιστές επέλεξαν να το χρησιμοποιήσουν ειδικά στην ανάπτυξη του Linux (Amidon, Jane. 2012).



2.3.5. Εγχειρίδιο εισαγωγής στην χρήση του Emacs Lisp

Ένα άλλο παράδειγμα είναι ο προγραμματισμός γλώσσας C, ο οποίος ήταν και αυτός περισσότερο αποδεκτός στην ανάπτυξη ανοικτού κώδικα. Το 1993 ο ρυθμός χρήσης της ανάπτυξης ανοικτού κώδικα έφθασε το 51%. Εκείνη την εποχή, το Make, το οποίο είναι το λογισμικό κοινής ωφέλειας για την κατασκευή των εκτελέσιμων

προγραμμάτων, ή βιβλιοθηκών στην ανάπτυξη UNIX, είχε μόνο έξι τοις εκατό ρυθμό χρήσης.



2.3.6. Βιβλίο εκμάθησης της γλώσσας C από τον δημιουργό της Dennis Ritchie

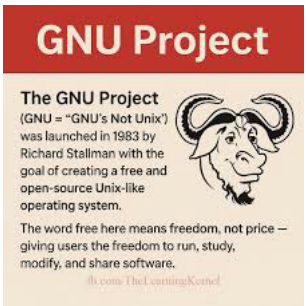
Προς το παρόν οι JAVA, C ++ και HTML αποτελούν τις δημοφιλέστερες γλώσσες προγραμματισμού ανοιχτού κώδικα. Επομένως, όλες αυτές οι γλώσσες προγραμματισμού ανοιχτού κώδικα επηρεάζουν τη νέα γλώσσα προγραμματισμού που θα εμφανιστεί στο μέλλον. Για παράδειγμα, το C επηρεάζει τα C ++, Python, JavaScript και Perl. Επιπρόσθετα, ο προγραμματισμός Java βασικά βοηθά το C# και

το Python να επηρεάσουν τον Ruby. Η χρήση των γλωσσών C και C ++ μειώνεται ενώ η γλώσσα JAVA εμφανίστηκε ξαφνικά και η χρήση της δεν έδειξε μεταβολή, παρά μετά από καιρό. Στην κατάσταση της γλώσσας XML, το μερίδιο χρήσης επίσης αυξήθηκε και είχε φτάσει στο υψηλότερο ποσοστό χρήσης (Coppinger, J. 2017).

2.4. To GNU project

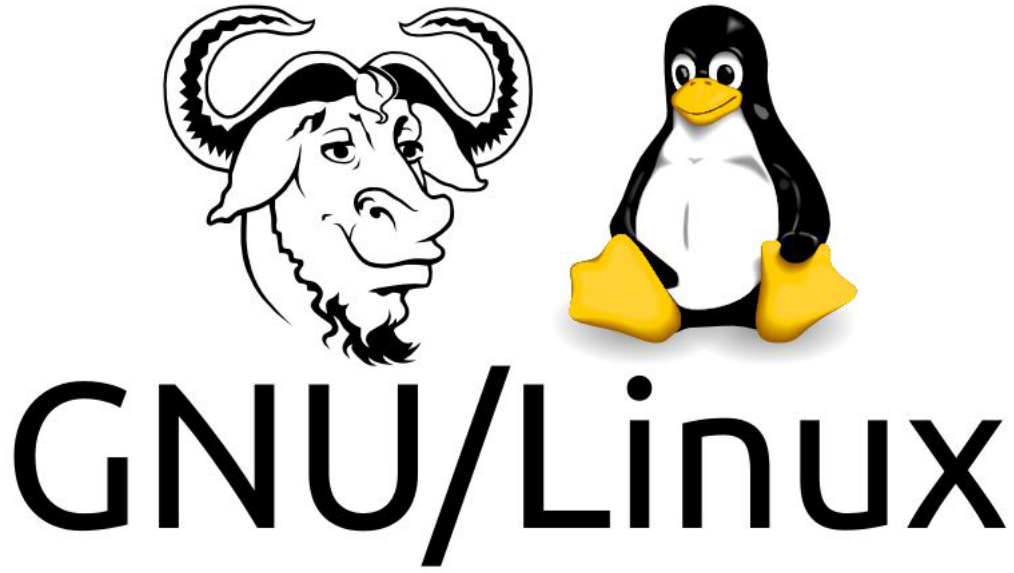
Το GNU project, είναι ένα έργο ελεύθερου λογισμικού, μαζικής συνεργασίας, που ανακοινώθηκε για πρώτη φορά στις 27 Σεπτεμβρίου 1983 από τον Richard Stallman στο MIT. Στόχος του είναι να παρέχει στους χρήστες ηλεκτρονικών υπολογιστών ελευθερία και έλεγχο στη χρήση των υπολογιστών και των υπολογιστικών τους συσκευών, αναπτύσσοντας και παρέχοντας λογισμικά που βασίζονται στα δικαιώματα ελευθερίας: οι χρήστες είναι ελεύθεροι να εκτελούν το λογισμικό, να το μοιράζονται (αντιγραφή, διανομή), να το μελετήσουν και να το τροποποιήσουν. Το λογισμικό GNU εγγυάται νομικά αυτά τα δικαιώματα ελευθερίας (μέσω της αδείας του) και ως εκ τούτου είναι ελεύθερο λογισμικό. Η χρήση της λέξης "ελεύθερο" πάντα να αναφέρεται στην ελευθερία (Herlitz, Jeff. 2012) .

2.4.1. Λογότυπο με την εννοιολογική σημασία του GNU



Προκειμένου να διασφαλιστεί ότι όλο το λογισμικό ενός υπολογιστή παρέχει στους χρήστες του όλα τα δικαιώματα ελευθερίας (χρήση, μετοχή, μελέτη, τροποποίηση), ακόμη και το πιο σημαντικό μέρος, το λειτουργικό σύστημα (συμπεριλαμβανομένων όλων των πολυάριθμων προγραμμάτων χρησιμότητάς του) ελεύθερο λογισμικό. Ο ιδρυτικός στόχος του έργου ήταν να δημιουργήσει ένα ελεύθερο λειτουργικό σύστημα και, ει δυνατόν, "όλα τα χρήσιμα που κανονικά υπάρχουν σε ένα σύστημα Unix, ώστε να μπορεί κανείς να συναντήσει χωρίς κανένα λογισμικό που δεν είναι ελεύθερο" (Babar, A. 2016).

Ο Stallman αποφάσισε να καλέσει αυτό το λειτουργικό σύστημα GNU (ένα αναδρομικό ακρωνύμιο που σημαίνει "GNU's not Unix"), βασίζοντας το σχεδιασμό του σε αυτό του Unix, ενός ιδιόκτητου λειτουργικού συστήματος. Η ανάπτυξη ξεκίνησε τον Ιανουάριο του 1984. Το 1991 εμφανίστηκε ο πυρήνας του Linux, που αναπτύχθηκε εκτός του έργου GNU από τον Linus Torvalds και τον Δεκέμβριο του 1992 διατέθηκε υπό την έκδοση 2 της Γενικής Δημόσιας Άδειας GNU. Σε συνδυασμό με τα βοηθητικά προγράμματα λειτουργικού συστήματος που έχουν ήδη αναπτυχθεί από το έργο GNU, επέτρεψε για το πρώτο λειτουργικό σύστημα το ελεύθερο λογισμικό, γνωστό και ως Linux. Το τρέχον έργο του project περιλαμβάνει την ανάπτυξη λογισμικού, τη δημιουργία ευαισθητοποίησης, την πολιτική εκστρατεία και την κοινή χρήση του νέου υλικού (3D Innovations. 2013).



2.4.2. GNU έναντι Linux

Ο Richard Stallman ανακοίνωσε την πρόθεσή του να ξεκινήσει την κωδικοποίηση του έργου GNU σε ένα μήνυμα Usenet τον Σεπτέμβριο του 1983. Όταν ξεκίνησε αρχικά το έργο GNU, είχε ένα πρόγραμμα επεξεργασίας κειμένου Emacs με την Lisp για τη σύνταξη εντολών επεξεργασίας, ένα πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής και ένας σύνδεσμος. Το σύστημα GNU απαιτούσε το δικό του μεταγλωττιστή C και τα εργαλεία του να είναι ελεύθερου λογισμικού, οπότε έπρεπε να αναπτυχθούν και μέχρι τον Ιούνιο του 1987, το έργο είχε συγκεντρώσει και ανέπτυξε ελεύθερο λογισμικό για έναν assembler, (GNU Emacs), και διάφορα βοηθητικά προγράμματα Unix (όπως ls, grep, awk, make και ld). Είχαν έναν αρχικό πυρήνα που χρειάζονταν περισσότερες ενημερώσεις. Το GNU ήταν σε θέση να χρησιμοποιηθεί για την ανάπτυξη του προγράμματος. Ο κύριος στόχος ήταν να δημιουργηθούν πολλές άλλες εφαρμογές που θα ήταν σαν το σύστημα Unix. Η GNU ήταν σε θέση να τρέξει τα προγράμματα Unix αλλά δεν ήταν πανομοιότυπα με αυτή. Το GNU ενσωμάτωσε μεγαλύτερα ονόματα αρχείων, τους αριθμούς έκδοσης αρχείων και ένα cras αδρανειακό σύστημα αρχείων (Amidon, Jane. 2012).

Το Μανιφέστο του GNU γράφτηκε για να λάβει υποστήριξη και συμμετοχή από άλλους για το έργο. Οι προγραμματιστές ενθαρρύνθηκαν να συμμετάσχουν σε οποιαδήποτε πτυχή του έργου που τους ενδιέφερε. Οι άνθρωποι θα μπορούσαν να δωρίσουν χρήματα, τμήματα ηλεκτρονικών υπολογιστών, ή ακόμα και τον δικό τους χρόνο για να γράψουν κώδικα και προγράμματα για το έργο. Η προέλευση και η ανάπτυξη των περισσότερων πτυχών του έργου GNU (και του ελεύθερου λογισμικού εν γένει) μοιράζονται σε μια λεπτομερή αφήγηση στο σύστημα βοήθειας του Emacs. (C-h g τρέχει την εντολή επεξεργασίας Emacs περιγράψτε-gnu-project.) Είναι το ίδιο λεπτομερές ιστορικό όπως και στην ιστοσελίδα τους (Coppinger, J. 2017).

Το μανιφέστο του GNU γράφτηκε από τον Richard Stallman για την υποστήριξη και συμμετοχή του στο έργο GNU. Στο μανιφέστο του GNU, ο Stallman απαρίθμησε τέσσερις ελευθερίες που είναι απαραίτητες για τους χρήστες λογισμικού: ελευθερία εκτέλεσης ενός προγράμματος για οποιονδήποτε σκοπό, ελευθερία μελέτης των μηχανικών του προγράμματος και τροποποίηση του, ελευθερία αναδιανομής αντιγράφων και ελευθερία βελτίωσης και αλλαγής τροποποιημένων εκδόσεων για το κοινό χρήση. Για να εφαρμοστούν αυτές οι ελευθερίες, οι χρήστες χρειάζονται πλήρη πρόσβαση στον κώδικα. Για να διασφαλίσει ότι ο κώδικας παρέμεινε ελεύθερος και θα παρέχονταν στο κοινό, ο Stallman δημιούργησε τη Γενική Άδεια Δημόσιας Χρήσης του GNU (GPL), η οποία επέτρεπε στο λογισμικό και στις μελλοντικές γενεές κώδικα που προέρχεται από αυτό να παραμείνει ελεύθερη για δημόσια χρήση (Babar, A. 2016).



2.4.3. Λογότυπο της αδειοδότησης GPL από τον Richard Stallman

2.5. Το FSF

Καθώς αυξανόταν το ενδιαφέρον για τη χρήση του Emacs, περισσότερα άτομα συμμετείχαν στο έργο GNU και κατά συνέπεια έφθασε η ώρα της αναζήτησης χρηματοδότησης. Έτσι το 1985 δημιουργήθηκε το Ίδρυμα Ελεύθερου Λογισμικού (FSF), ένα αφοσιωμένο φιλανθρωπικό ίδρυμα για την ανάπτυξη του ελεύθερου λογισμικού. Το FSF ανέλαβε επίσης την επιχείρηση διανομής ταινιών Emacs, αργότερα επέκτεινε αυτό με την προσθήκη άλλου ελεύθερου λογισμικού (GNU και μη GNU) στην κασέτα και με την πώληση δωρεάν εγχειριδίων. Το μεγαλύτερο μέρος του εισοδήματος του FSF προήλθε από την πώληση αντιγράφων του ελεύθερου λογισμικού και άλλων σχετικών υπηρεσιών (CD-ROM του πηγαίου κώδικα, CD-ROM με δυαδικά αρχεία, ωραία τυπωμένα εγχειρίδια, όλα με την ελευθερία αναδιανομής και τροποποίησης) και Deluxe Διανομές (διανομές για τις οποίες δημιουργήθηκε ολόκληρη συλλογή του λογισμικού για την επιλογή της πλατφόρμας του πελάτη) (Hendrix, John Shannon. 2013).



2.5.1. Ίδρυμα Ελεύθερου Λογισμικού (FSF) 40η επέτειος

Σήμερα το FSF εξακολουθεί να πωλεί εγχειρίδια και άλλα εργαλεία, αλλά παίρνει το μεγαλύτερο μέρος της χρηματοδότησής του από τα τέλη των μελών. Οι υπάλληλοι του Ιδρύματος Ελεύθερου Λογισμικού έχουν συντάξει και διατηρούν πολλά πακέτα λογισμικού GNU. Δύο αξιοσημείωτες είναι η βιβλιοθήκη C και το κέλυφος. Η βιβλιοθήκη GNU C είναι αυτή που κάθε πρόγραμμα που τρέχει σε ένα σύστημα GNU / Linux χρησιμοποιείται ούτως ώστε να επικοινωνεί με το Linux. Αναπτύχθηκε από ένα μέλος του προσωπικού του Ιδρύματος Ελεύθερου Λογισμικού, Roland McGrath. Το κέλυφος που χρησιμοποιείται στα περισσότερα συστήματα GNU / Linux είναι το BASH, το Bourne Again Shell, 8 το οποίο αναπτύχθηκε από τον υπάλληλο του FSF Brian Fox (Coppinger, J. 2017).

2.6. Εταιρία Ελεύθερου Λογισμικού και οι Στόχοι

Το FSF διατηρεί έναν κατάλογο με "έργα υψηλής προτεραιότητας" στα οποία το Ίδρυμα ισχυρίζεται ότι «υπάρχει ζωτική ανάγκη να προσελκύσουμε την προσοχή της κοινότητας του ελεύθερου λογισμικού». Το FSF θεωρεί αυτά τα έργα "σημαντικά επειδή οι χρήστες ηλεκτρονικών υπολογιστών συνεχώς γοητεύονται στη χρήση μη-ελεύθερου λογισμικού, επειδή δεν υπάρχει επαρκής δωρεάν αντικατάσταση". Οι τρέχουσες εργασίες υψηλής προτεραιότητας περιλαμβάνουν ιδιόκτητο υλικολογισμικό αντίστροφης μηχανικής. Αναστρέψιμη αποσφαλμάτωση στο GNU Debugger. Ανάπτυξη λογισμικού αυτόματης μεταγραφής και επεξεργασίας βίντεο, Coreboot, προγράμματα οδήγησης για δρομολογητές δικτύου και δημιουργία αντικαταστάσεων για το Skype, το Google Earth, τις βιβλιοθήκες OpenDWG, το BitTorrent Sync και τα έντυπα Oracle (3D Innovations. 2013).

Προηγούμενα έργα που επισημάνθηκαν ως έργα που χρειάζονταν περιλαμβάνουν τις εφαρμογές Free Java, το GNU Classpath και το GNU Compiler για Java, οι οποίες διασφαλίζουν τη συμβατότητα για το τμήμα Java του OpenOffice.org και το περιβάλλον εργασίας GNOME. Η προσπάθεια έχει επικριθεί από τον Michael Larabel είτε για να μην υποκινήσει την ενεργό ανάπτυξη ούτε για να επιβραδύνει το έργο που επιτελείται, ακόμα και μετά την προσθήκη ορισμένων σχεδίων στον κατάλογο (Amidon, Jane. 2012).

2.7. Πλεονεκτήματα Ελεύθερου Λογισμικού

Το ελεύθερο λογισμικό είναι φθηνότερο από τα εμπορικά προϊόντα που διατίθενται στο εμπόριο. Σύμφωνα με μελέτες, το λογισμικό ανοιχτού κώδικα συλλογικά βοηθά τους ιδιοκτήτες επιχειρήσεων να εξοικονομήσουν περίπου 60 δισεκατομμύρια δολάρια ετησίως. Αυτό μπορεί να φαίνεται απίστευτο, αλλά δεν προκαλεί έκπληξη το γεγονός ότι αυτά τα προγράμματα αναπτύσσονται για να υπάρχει πρόσβαση η οποία θα παρέχεται σε όλους (ειδικά σε εκείνους που δεν έχουν την οικονομική δυνατότητα να αγοράσουν εμπορικά προϊόντα). Σημασία έχει πώς προσφέρονται συνήθως δωρεάν και δεν απαιτούν από τους ανθρώπους να πληρώσουν για οποιοδήποτε επιπλέον αντίγραφο κατεβάσουν. Δεδομένου ότι πολλά από αυτά τα προγράμματα δημιουργούνται για να λειτουργούν με σχεδόν οποιοδήποτε τύπο υπολογιστή, μπορούν επίσης να βοηθήσουν να παραταθεί η ζωή του παλιού υλικού και να αποφευχθεί η ανάγκη αντικατάστασης τους κάθε τόσο (Babar, A. 2016).

Δημιουργήθηκε από επιδέξιους και ταλαντούχους ανθρώπους. Μεγάλες και καθιερωμένες εταιρείες λογισμικού έχουν την οικονομική δυνατότητα να προσλάβουν τα καλύτερα ταλέντα στην επιχείρηση για να δημιουργήσουν τα προϊόντα τους. Εξαιτίας αυτού, πολλοί άνθρωποι επιλέγουν να αγοράσουν προγράμματα ηλεκτρονικών υπολογιστών από αυτές τις εταιρείες επειδή νομίζουν ότι θα έχουν μεγάλη αξία για τα χρήματά τους με αυτόν τον τρόπο. Το γεγονός ότι οι μεγάλες εταιρείες προσλαμβάνουν έμπειρους, πλήρως εκπαιδευμένους και άριστα καταρτισμένους ανθρώπους είναι αλήθεια. Αλλά αυτό που πρέπει να γνωρίζουν οι καταναλωτές είναι αυτό: δεν ενδιαφέρονται όλοι οι προγραμματιστές λογισμικού βαθιά για τα χρήματα. Σίγουρα, οι περισσότεροι από αυτούς έχουν δουλειά για να έχουν ένα σταθερό εισόδημα και να είναι σε θέση να υποστηρίξουν οικονομικά τον εαυτό τους και την οικογένειά τους (Hendrix, John Shannon. 2013). Ωστόσο, πολλοί

από αυτούς τους εμπειρογνώμονες δεν βασίζουν μόνο την αξία τους στον μισθό που κερδίζουν. Αντίθετα, προσπαθούν να οικοδομήσουν ένα πρόγραμμα που θα κερδίσει το θαυμασμό της κοινωνίας τους και ελπίζουν ότι θα κάνει τη διαφορά στον κόσμο.

Αυτός είναι ο λόγος που πολλοί προγραμματιστές λογισμικού μετατρέπουν σε προϊόντα ανοιχτού κώδικα ως σημείο πώλησης για τις ιδέες και τη δημιουργικότητά τους. Με αυτόν τον τρόπο, δεν θα περιορίζονται από τους άκαμπτους κανόνες του εταιρικού κόσμου και θα έχουν την ελευθερία να πειραματιστούν και να παρουσιάσουν προγράμματα υψηλής ποιότητας. Αυτά, με τη σειρά τους, ωφελούν τους καταναλωτές, αφού έχουν πρόσβαση σε λογισμικό παγκόσμιας κλάσης και τελευταίας τεχνολογίας, χωρίς να χρειαστεί να πληρώσουν πάρα πολλά.

Είναι εξαιρετικά αξιόπιστα. Υπάρχουν δύο κύριοι λόγοι για τους οποίους το λογισμικό ανοιχτού κώδικα είναι αξιόπιστο. Πρώτα απ 'όλα, αναπτύσσονται κυρίως από έμπειρους και ταλαντούχους ειδικούς που κάνουν ό, τι καλύτερο μπορούν να δημιουργήσουν προγράμματα υψηλής ποιότητας. Δεύτερον, έχουν επεξεργασθεί από δεκάδες ή εκατοντάδες ανθρώπους, πράγμα που σημαίνει ότι υπάρχουν πολλά μάτια που μπορούν να παρακολουθήσουν για την παρουσία σφαλμάτων και πολλά ζευγάρια χεριών που μπορούν να διορθώσουν αυτά τα ελαττώματα μέσα στο συντομότερο χρονικό διάστημα. Και οι δύο αυτοί παράγοντες οδηγούν σε προϊόντα που έχουν άριστη ποιότητα και χρήσιμα χαρακτηριστικά και εκτελούν τα περισσότερα (αν όχι όλα) του χρόνου (Babar, A. 2016).

Υπάρχει ευελιξία. Δεδομένου ότι δεν δεν υπάρχει σύνδεση με ένα ιδιόκτητο προϊόν, δεν χρειάζεται να υπάρχει συμμόρφωση με μια συγκεκριμένη αρχιτεκτονική

πληροφορικής που ενδέχεται να απαιτεί την αναβάθμιση του λογισμικού και του υλικού (Amidon, Jane. 2012).

2.8. Μειονεκτήματα Ελεύθερου Λογισμικού

Καταρχάς, είναι ευάλωτα σε κακόβουλους χρήστες. Πολλοί άνθρωποι έχουν πρόσβαση στον πηγαίο κώδικα του λογισμικού ανοικτού κώδικα, αλλά δεν έχουν όλοι καλή πρόθεση. Ενώ πολλοί άνθρωποι χρησιμοποιούν την πρόσβασή τους για ελαττώματα σημείων και βελτιώνουν το πρόγραμμα, άλλοι χρησιμοποιούν αυτό το προνόμιο για να εκμεταλλευτούν τα τρωτά σημεία του προϊόντος και να δημιουργήσουν σφάλματα που μπορούν να μολύνουν υλικό, να κλέψουν ταυτότητες ή απλώς να ενοχλήσουν άλλους χρήστες. Αυτά συμβαίνουν σπάνια με το εμπορικά παραγόμενο λογισμικό, καθώς οι εταιρείες που τις κατασκευάζουν έχουν αυστηρές διαδικασίες ελέγχου ποιότητας και εξασφαλίζουν ότι το πρόγραμμα είναι σχεδόν τέλειο όταν κυκλοφορήσει στην αγορά (Babar, A. 2016).

Δεύτερον, δεν μπορεί να είναι τόσο φιλική προς το χρήστη όσο και οι εμπορικές εκδόσεις. Αυτό δεν ισχύει για όλα τα λογισμικά ανοικτού κώδικα, καθώς πολλά από αυτά (όπως το LibreOffice, το Mozilla Firefox και το λειτουργικό σύστημα Android) είναι απίστευτα εύκολα στη χρήση. Ωστόσο, υπάρχουν πολλά προγράμματα που δημιουργούνται κυρίως για να ικανοποιήσουν τις επιθυμίες του προγραμματιστή και να φέρουν τις ιδέες του στη ζωή. Ως αποτέλεσμα, δεν δίνεται μεγάλη προσοχή στο περιβάλλον χρήστη του λογισμικού, καθιστώντας δύσκολη τη χρήση του ειδικά για εκείνους που δεν είναι πραγματικά γνώστες της τεχνολογίας (Indovance.com. 2015).

Τρίτον, δεν υπάρχει εκτεταμένη υποστήριξη. Εκείνοι που ευνοούν τα εμπορικά παραγόμενα προγράμματα λένε ότι αυτό το λογισμικό τους διασφαλίζει ηρεμία. Εξάλλου, αφού ξέρουν ακριβώς ποιος σχεδίασε, δημιούργησε και διανέμει το προϊόν, έχουν σαφή εικόνα του ποιος μπορεί να θεωρηθεί υπεύθυνος εάν το πρόγραμμα δεν λειτουργήσει σωστά ή προκαλέσει ζημιά στο υλικό του. Αυτό δεν ισχύει ακριβώς για το λογισμικό ανοιχτού κώδικα. Δεδομένου ότι αναπτύχθηκε από πολλούς ανθρώπους οι χρήστες δεν έχουν ακριβώς ένα συγκεκριμένο άτομο, ή εταιρεία που μπορούν να υποδείξουν ως υπεύθυνους (3D Innovations, 2013).

2.9. Διαφορές Λογισμικού Ανοικτού και Κλειστού Κώδικα

Πολιτική Τιμών

Η ανοιχτή πηγή αναφέρεται συχνά ως δωρεάν λογισμικό κόστους. Μπορεί ωστόσο να έχει κόστος για πρόσθετα όπως βοήθεια, πρόσθετες υπηρεσίες, ή πρόσθετη λειτουργικότητα. Έτσι, μπορεί να πληρωθεί για μια υπηρεσία με το OSS. Το κλειστό λογισμικό πηγής είναι συνήθως λογισμικό πληρωμής. Το κόστος μπορεί να ποικίλει ανάλογα με την πολυπλοκότητα του λογισμικού, ενώ η τιμή μπορεί να είναι υψηλότερη, να είναι ένα καλύτερο προϊόν, πλήρης υποστήριξη, λειτουργικότητα και καινοτομία. Ωστόσο, οι περισσότερες εταιρείες παρέχουν δωρεάν δοκιμές για να πείσουν τον αγοραστή ότι το λογισμικό τους είναι αρεστό (Coppinger, J. 2017).

Ασφάλεια

Το ζήτημα της ασφάλειας είναι πολύ αμφιλεγόμενο, καθώς κάθε λογισμικό χαρακτηρίζεται σαν τις δύο όψεις ενός νομίσματος. Ο κώδικας του λογισμικού ανοιχτού κώδικα μπορεί να προβληθεί, να μοιραστεί και να τροποποιηθεί από την κοινότητα, πράγμα που σημαίνει ότι ο καθένας μπορεί να διορθώσει, τα σφάλματα διορθώνονται γρήγορα και ο κώδικας ελέγχεται διεξοδικά μετά από κάθε απελευθέρωση. Ωστόσο, λόγω της διαθεσιμότητας, ο πηγαίος κώδικας είναι ανοιχτός για χάκερς που μπορούν να ασκήσουν τις ικανότητες τους (Babar, A. 2016).

Αντίθετα, το λογισμικό κλειστής πηγής μπορεί να καθοριστεί μόνο από έναν προμηθευτή. Εάν κάτι πάει στραβά με το λογισμικό, οι πελάτες στέλνουν ένα αίτημα και περιμένουν την απάντηση από την ομάδα υποστήριξης. Η επίλυση του προβλήματος μπορεί να διαρκέσει πολύ περισσότερο από ό, τι σε σχέση με την OSC. Όταν πρόκειται για την επιλογή του πιο ασφαλούς λογισμικού, η απάντηση είναι ότι καθένας από αυτούς έχει τα υπέρ και τα κατά. Έτσι είναι συχνά μια πρόκληση για τις επιχειρήσεις που εργάζονται σε συγκεκριμένο κλάδο (Babar, A. 2016).

Ποιότητα υποστήριξης

Συγκρίνοντας την υποστήριξη λογισμικού ανοικτού κώδικα και κλειστού κώδικα, είναι προφανές ότι το CSS κυριαρχεί στην περίπτωση αυτή. Το κόστος για αυτό περιλαμβάνει την επιλογή επικοινωνίας με την υποστήριξη και παραλαβή σε μία εργάσιμη ημέρα στις περισσότερες περιπτώσεις. Η απάντηση είναι καλά οργανωμένη και τεκμηριωμένη. Για λογισμικό ανοιχτού κώδικα, αυτή η επιλογή δεν παρέχεται. Οι μόνες επιλογές υποστήριξης είναι φόρουμ, χρήσιμα άρθρα και εμπειρογνώμονες.

Ωστόσο, δεν προκαλεί έκπληξη το γεγονός ότι η χρήση τέτοιου είδους υπηρεσίας δεν θα λάβει υψηλό επίπεδο ανταπόκρισης (Hendrix, John Shannon. 2013).

Διαθεσιμότητα πηγαίου κώδικα

Το λογισμικό ανοιχτού κώδικα παρέχει τη δυνατότητα αλλαγής του πηγαίου κώδικα χωρίς περιορισμούς. Οι μεμονωμένοι χρήστες μπορούν να αναπτύξουν αυτό που θέλουν και να επωφεληθούν από την καινοτομία που αναπτύσσουν άλλοι μέσα στην κοινότητα των χρηστών. Καθώς ο πηγαίος κώδικας είναι εύκολα προσβάσιμος, επιτρέπει στους προγραμματιστές λογισμικού να βελτιώσουν τα ήδη υπάρχοντα προγράμματα. Το λογισμικό πηγαίου κώδικα είναι πιο περιορισμένο από λογισμικό ανοιχτού κώδικα, επειδή ο πηγαίος κώδικας δεν μπορεί να αλλάξει, ή να προβληθεί. Ωστόσο, ένας τέτοιος περιορισμός είναι αυτό που μπορεί να συμβάλει στην ασφάλεια και την αξιοπιστία του CSS (Amidon, Jane. 2012).

Ευχρηστία

Η χρηστικότητα είναι ένα επίπονο θέμα του λογισμικού ανοιχτού κώδικα. Οι οδηγοί χρηστών γράφονται για τους προγραμματιστές και όχι για τους χρήστες που θέλουν να κάνουν. Επίσης, αυτά τα εγχειρίδια δεν συμμορφώνονται με τα πρότυπα και τη δομή. Για κλειστού κώδικα, η χρηστικότητα του λογισμικού είναι ένα από τα πλεονεκτήματα. Η τεκμηρίωση είναι συνήθως σωστά γραμμένη και περιέχει λεπτομερείς οδηγίες (Babar, A. 2016).

2.10. Άδειες Χρήσης Λογισμικού

Μια άδεια χρήσης λογισμικού είναι ένα νομικό μέσο (συνήθως μέσω του δικαίου των συμβάσεων με, ή χωρίς έντυπο υλικό) που διέπει τη χρήση, ή την ανακατανομή του λογισμικού. Σύμφωνα με το νόμο περί πνευματικών δικαιωμάτων των Ηνωμένων Πολιτειών, όλο το λογισμικό προστατεύεται από πνευματικά δικαιώματα, τόσο σε πηγαίο κώδικα όσο και σε μορφές κωδικών αντικειμένων. Η μόνη εξαίρεση είναι το λογισμικό στον δημόσιο τομέα. Μια τυπική άδεια χρήσης παρέχει στον δικαιοδόχο, συνήθως έναν τελικό χρήστη, άδεια χρήσης ενός, ή περισσοτέρων αντιγράφων του λογισμικού με τρόπους όπου μια τέτοια χρήση θα μπορούσε ενδεχομένως να αποτελεί παραβίαση πνευματικών δικαιωμάτων των αποκλειστικών δικαιωμάτων πνευματικής ιδιοκτησίας του κατόχου του λογισμικού (3D Innovations. 2013).

Το λογισμικό μπορεί να κατηγοριοποιηθεί ανάλογα με τον τύπο της άδειας. Δύο κοινές κατηγορίες για το λογισμικό βάσει του νόμου περί πνευματικών δικαιωμάτων και συνεπώς με τις άδειες που χορηγούν τα ειδικά δικαιώματα του δικαιοπάροχου είναι ιδιόκτητο λογισμικό και ελεύθερο λογισμικό ανοικτού κώδικα (FOSS). Η διακριτή εννοιολογική διαφορά μεταξύ των δύο είναι η παραχώρηση δικαιωμάτων τροποποίησης και επαναχρησιμοποίησης ενός προϊόντος λογισμικού που αποκτάται από έναν πελάτη: Το λογισμικό FOSS εκχωρεί ταυτόχρονα δικαιώματα στον πελάτη και συνεπώς συνδέει τον τροποποιημένο πηγαίο κώδικα με το λογισμικό ("ανοικτή πηγή"), ενώ το ιδιόκτητο λογισμικό συνήθως δεν παρέχει άδεια χρήσης αυτών των δικαιωμάτων και επομένως διατηρεί τον πηγαίο κώδικα κρυφό ("κλειστή πηγή").

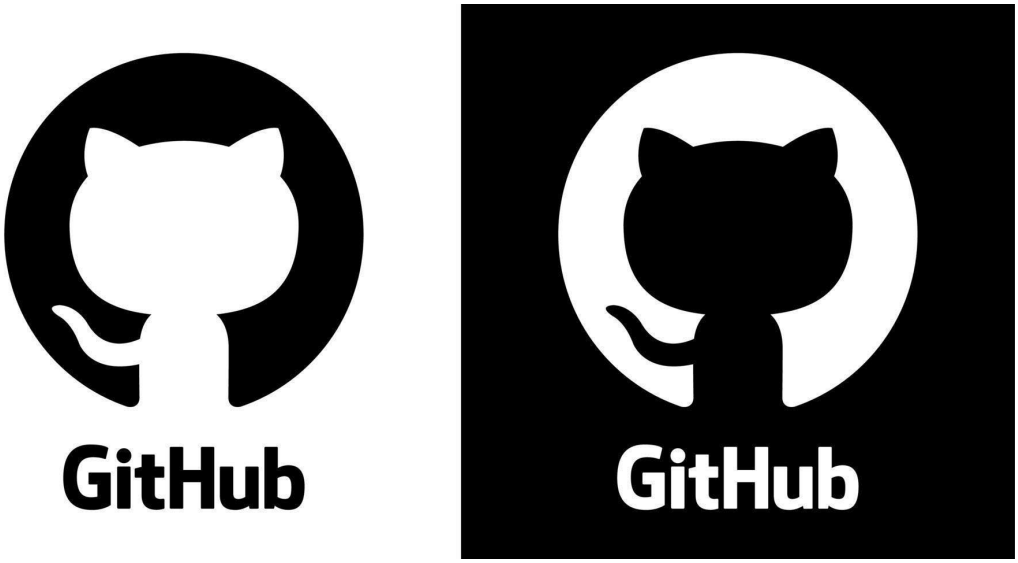


2.10.1. Λογότυπο λογισμικού αξιολόγησης ελεύθερου και ανοικτού κώδικα (FOSS)

Εκτός από τη χορήγηση δικαιωμάτων και την επιβολή περιορισμών στη χρήση λογισμικού που προστατεύεται από πνευματικά δικαιώματα, οι άδειες χρήσης λογισμικού περιέχουν συνήθως διατάξεις που κατανέμουν ευθύνη και ευθύνη μεταξύ των συμβαλλομένων που συνάπτουν τη σύμβαση άδειας χρήσης. Σε επιχειρηματικές και εμπορικές συναλλαγές λογισμικού, αυτοί οι όροι περιλαμβάνουν συχνά περιορισμούς ευθύνης, εγγυήσεων και αποποίησης ευθυνών και αποζημίωση αν το λογισμικό παραβιάζει τα δικαιώματα πνευματικής ιδιοκτησίας οποιουδήποτε (Amidon, Jane. 2012).

Το μη εγκεκριμένο λογισμικό εκτός του πεδίου προστασίας της πνευματικής ιδιοκτησίας είναι, είτε λογισμικό δημοσίου τομέα (PD), είτε λογισμικό το οποίο είναι μη διανεμημένο, μη εγκεκριμένο και χειρισμένο ως εσωτερικό εμπορικό μυστικό της επιχείρησης. Σε αντίθεση με τις κοινές πεποιθήσεις, το διανεμόμενο μη εγκεκριμένο λογισμικό (δεν είναι δημόσιος τομέας) προστατεύεται πλήρως από πνευματικά δικαιώματα και ως εκ τούτου νομικά αχρησιμοποίητο (δεδομένου ότι κανένα

δικαίωμα χρήσης δεν χορηγείται με άδεια) μέχρι να περάσει σε δημόσιο τομέα μετά τη λήξη της προθεσμίας. Παραδείγματα αυτού είναι οι μη εξουσιοδοτημένες διαρροές λογισμικού, ή τα έργα λογισμικού που τοποθετούνται σε δημόσιους χώρους αποθήκευσης λογισμικού όπως το GitHub χωρίς συγκεκριμένη άδεια. Δεδομένου ότι σε ορισμένες περιοχές του διεθνούς δικαίου (για παράδειγμα ο νόμος της Γερμανίας) είναι προβληματική η απόκτηση λογισμικού στο δημόσιο τομέα (πριν από την επίτευξη του δικαιώματος πνευματικής ιδιοκτησίας), υπάρχουν επίσης άδειες που χορηγούν δικαιώματα όπως το CC0 ή το WTFPL (Hendrix, John Shannon. 2013).



2.10.2. Λογότυπο του λογισμικού Github

Στις Ηνωμένες Πολιτείες, το άρθρο 117 του νόμου περί πνευματικής ιδιοκτησίας παρέχει στον κάτοχο ενός συγκεκριμένου αντιγράφου του λογισμικού το ρητό δικαίωμα χρήσης του λογισμικού με υπολογιστή, ακόμη και αν η χρήση του λογισμικού με έναν υπολογιστή απαιτεί την πραγματοποίηση παρεπόμενων αντιγράφων ή προσαρμογών που διαφορετικά θα μπορούσαν ενδεχομένως να

αποτελούν παραβίαση πνευματικών δικαιωμάτων). Ως εκ τούτου, ο κάτοχος ενός αντιγράφου λογισμικού υπολογιστή έχει το νόμιμο δικαίωμα να χρησιμοποιεί αυτό το αντίγραφο του λογισμικού. Συνεπώς, αν ο τελικός χρήστης του λογισμικού είναι ο κάτοχος του αντίστοιχου αντιγράφου, τότε ο τελικός χρήστης μπορεί νόμιμα να χρησιμοποιήσει το λογισμικό χωρίς άδεια από τον εκδότη λογισμικού (Coppinger, J. 2017).

Καθώς πολλές "άδειες" ιδιοκτησίας απαριθμούν μόνο τα δικαιώματα που ο χρήστης έχει ήδη κάτω από 17 U.S.C. § 117 και αν και διακηρύσσονται για να πάρουν τα δικαιώματα μακριά από το χρήστη, αυτές οι συμβάσεις μπορεί να στερούνται προσοχής. Οι άδειες ιδιοκτησιακού λογισμικού συχνά διακηρύσσονται για να δίνουν στους εκδότες λογισμικού μεγαλύτερο έλεγχο στον τρόπο με τον οποίο το λογισμικό τους χρησιμοποιείται, διατηρώντας την ιδιοκτησία κάθε αντιγράφου λογισμικού στον εκδότη λογισμικού. Με τον τρόπο αυτό η ενότητα 117 δεν ισχύει για τον τελικό χρήστη και ο εκδότης λογισμικού μπορεί να υποχρεώσει τον τελικό χρήστη να αποδεχθεί όλους τους όρους της σύμβασης άδειας χρήσης, πολλοί από τους οποίους ενδέχεται να είναι πιο περιοριστικοί από ό,τι ο νόμος περί πνευματικών δικαιωμάτων. Η μορφή της σχέσης καθορίζει εάν πρόκειται για μίσθωση, ή αγορά, για παράδειγμα UMG εναντίον Augusto, ή Vernor vs. Autodesk, Inc (Babar, A. 2016).



2.10.3. Λογότυπο της πλατφόρμας πώλησης, διακίνησης και χρήσης βιντεοπαιχνιδιών Steam

Η ιδιοκτησία ψηφιακών αγαθών, όπως οι εφαρμογές λογισμικού και τα βιντεοπαιχνίδια, αμφισβητείται από τις "αδειοδοτημένες, μη πωλημένες" ΑΔΕΚ των ψηφιακών διανομέων, όπως η Steam. Στην Ευρωπαϊκή Ένωση το Δικαστήριο των Ευρωπαϊκών Κοινοτήτων έκρινε ότι ο κάτοχος των πνευματικών δικαιωμάτων δεν μπορεί να αντιταχθεί στη μεταπώληση ενός ψηφιακού λογισμικού που πωλείται, σύμφωνα με τον κανόνα περί εξάντλησης του δικαιώματος πνευματικής ιδιοκτησίας κατά την πρώτη πώληση, καθώς μεταβιβάζεται η κυριότητα και επομένως αμφισβητεί την EULA.Η βασισμένη στην Ελβετία εταιρεία UsedSoft καινοτομούσε τη μεταπώληση επιχειρηματικού λογισμικού και αγωνίστηκε για αυτό το δικαίωμα στο δικαστήριο. Στην Ευρώπη η οδηγία 2009/24 / ΕΚ της ΕΕ επιτρέπει ρητά την εμπορία χρησιμοποιημένων προγραμμάτων ηλεκτρονικών υπολογιστών (Amidon, Jane. 2012).

2.11. Άδειες Χρήσης Ελεύθερου Λογισμικού

Ενώ το λογισμικό ανοιχτού κώδικα παρέχει πολλά πολύτιμα οφέλη στις επιχειρήσεις και την κοινότητα ανάπτυξης λογισμικού, οι άδειες χρήσης λογισμικού ανοιχτού κώδικα συχνά περιέχουν αυστηρές απαιτήσεις. Όταν οι εταιρείες χρησιμοποιούν

λογισμικό ανοιχτού κώδικα στα προϊόντα τους χωρίς να κατανοούν τους όρους των εφαρμοστέων αδειών, ενδέχεται να προκληθούν κατά λάθος κίνδυνοι που σχετίζονται με την προστασία των πολύτιμων δικαιωμάτων της πνευματικής ιδιοκτησίας και ως εκ τούτου να αμβλύνουν την αξία των προϊόντων λογισμικού και των επιχειρήσεων τους (Indovance.com. 2015).

Η ευρεία χρήση του λογισμικού ανοιχτού κώδικα υποδηλώνει ότι οι πρακτικές επιπτώσεις του είναι ευρέως κατανοητές και εκτιμημένες. Δυστυχώς, οι νομικές συνέπειες της χρήσης λογισμικού ανοιχτού κώδικα δεν είναι ακόμη ευρέως γνωστές, ή κατανοητές. Οι άδειες λογισμικού ανοικτού κώδικα εμπίπτουν συνήθως σε μία από τις δύο ευρείες κατηγορίες: άδειες (είτε μη) και περιοριστικές (είτε copyleft) άδειες (3D Innovations. 2013).

Επιτρεπόμενες άδειες

Οι επιτρεπτές άδειες χρήσης επιτρέπουν κατά κανόνα την ελεύθερη και απεριόριστη χρήση του εγκεκριμένου λογισμικού, εφόσον ο χρήστης συμμορφώνεται με ορισμένες ελάχιστες απαιτήσεις, οι οποίες περιλαμβάνουν συνήθως την απόδοση της κατανομής όταν ο χρήστης χρησιμοποιεί τον κώδικα του δημιουργού στο δικό του λογισμικό, συμπεριλαμβάνοντας μια ειδοποίηση πνευματικών δικαιωμάτων τον πηγαίο κώδικα της εργασίας. Ορισμένες κοινώς χρησιμοποιούμενες άδειες είναι η άδεια MIT, η άδεια χρήσης Apache 2.0 και η άδεια BSD. Για επιχειρήσεις που ενδιαφέρονται να πουλήσουν ιδιόκτητο λογισμικό, οι άδειες άδειας γενικά θεωρούνται χαμηλού κινδύνου, υψηλών ανταμοιβών (Babar, A. 2016).

Περιοριστικές άδειες χρήσης

Όπως υποδηλώνει η ονομασία, οι περιοριστικές (copyleft) άδειες περιέχουν πολύ πιο επαχθείς απαιτήσεις από τις επιτρεπόμενες άδειες και παρουσιάζουν μεγαλύτερο βαθμό κινδύνου για τις εταιρείες που επιθυμούν να ενσωματώσουν κωδικό με άδεια χρήσης copyleft στα προϊόντα τους. Ορισμένες συνηθισμένες περιοριστικές άδειες είναι οι GPL (εκδόσεις 2 και 3), το AffPLER GPL (AGPL), το Lesser GPL (LGPL) και το Mozilla Public License 2.0 (MPL) . Αξίζει να σημειωθεί, ωστόσο, ότι το LGPL και το MPL θεωρούνται "αδύναμες" άδειες copyleft επειδή οι απαιτήσεις τους είναι λιγότερο περιοριστικές από τις απαιτήσεις της GPL και της AGPL (Jones, Stuart Mackenzie. 2008) .



2.11.1. Copyleft: Η εναλλακτική μέθοδος της ελεύθερης αδειοδότησης

Οι άδειες Copyleft επιδιώκουν να πραγματοποιήσουν κάποια εκδοχή της ριζοσπαστικής άποψης του FSF για την ελευθερία του λογισμικού, επικαλούμενες το νόμο περί πνευματικών δικαιωμάτων αλλά υπονομεύοντας τους παραδοσιακούς του στόχους. Ενώ τα πνευματικά δικαιώματα παραδοσιακά ασχολούνται με την

προστασία του δικαιώματος του δημιουργού να αποκλείει άλλους από την αντιγραφή, διανομή, ή τροποποίηση ενός έργου, το copyleft επιδιώκει να ενισχύσει τα δικαιώματα των χρηστών απαγορεύοντας την επιβολή ορίων στο δικαίωμα αντιγραφής, διανομής ή τροποποίησης ενός έργου. Στην πράξη, οι άδειες copyleft απαιτούν την άδεια χρήσης του έργου, καθώς και κάθε εργασία που βασίζεται στην άδεια χρήσης (που αναφέρεται παρακάτω), να συνεχίσει να διαθέτει άδεια με την ίδια άδεια χρήσης του copyleft. Για παράδειγμα, όλες οι εργασίες που ενσωματώνουν, ή βασίζονται με άλλο τρόπο σε λογισμικό που έχει λάβει άδεια χρήσης στο πλαίσιο της GPL πρέπει επίσης να έχουν άδεια χρήσης στο πλαίσιο της GPL. Αυτό έχει περιγραφεί ως η "ιογενής" φύση του copyleft επειδή ο copyleft-licensed κώδικας "μολύνει" κάθε κώδικα που αγγίζει. Επιπλέον μια εταιρεία που ενσωματώνει κώδικα copyleft στο προϊόν της πρέπει να παρέχει τον πλήρη πηγαίο κώδικα για το προϊόν λογισμικού σε όλους τους μεταγενέστερους αποδέκτες (Δηλ. Σε οποιονδήποτε στον οποίο η εταιρεία διανέμει εκτελέσιμο αντίγραφο του λογισμικού της) και δεν μπορεί να επιβάλει περιορισμούς δικαιώματα των επόμενων παραληπτών να αντιγράψουν, ή να τροποποιήσουν το πρόγραμμα (Amidon, Jane. 2012).

Ο πρακτικός αντίκτυπος της χρήσης του λογισμικού copyleft από μια εταιρεία στο ιδιόκτητο προϊόν της, χωρίς να λαμβάνεται δεόντως υπόψη ο κίνδυνος που συνδέεται με αυτό, μπορεί να είναι ασυμβίβαστος. Σε περίπτωση που ενεργοποιηθούν οι υποχρεώσεις copyleft και μια εταιρεία υποχρεούται να αποδεσμεύσει τον πηγαίο κώδικα που βασίζεται στο ιδιόκτητο προϊόν λογισμικού της, μπορεί να είναι δύσκολο να εμπορευματοποιηθεί το προϊόν της και η αξία της επιχείρησής της μπορεί να πέσει. Είναι σημαντικό ότι η υποχρέωση copyleft να παρέχει τον πηγαίο κώδικα ενός

προγράμματος ενεργοποιείται μόνο όταν έχει δημιουργηθεί ένα παράγωγο έργο του έργου copyleft και έχει παραχωρηθεί το παράγωγο έργο (Babar, A. 2016).

2.12. Τι είναι εκπαιδευτικό Λογισμικό

Η βελτίωση της απόδοσης των πραγματικών λογισμικών είναι γενικά ένα πολύ δύσκολο έργο. Ακόμα κι αν η ανώτερη διοίκηση της εκάστοτε εταιρίας είναι διατεθειμένη να υποστηρίξει τις αλλαγές της διαδικασίας και να πληρώσει για δαπανηρές συμβουλές, κατάρτιση και εργαλεία, οι προγραμματιστές με εχθρικό hacking θα αντισταθούν στις αλλαγές. Μπορούν να αναγνωρίσουν την ανάγκη για καλύτερες διαδικασίες, αλλά γενικά φοβούνται ότι ενδέχεται να καθυστερήσουν στις παραδόσεις προϊόντων και να θέσουν σε κίνδυνο τη δουλειά τους. Ως εκ τούτου, η διδασκαλία των μαθητών να ακολουθήσουν συγκεκριμένες διαδικασίες ανάπτυξης λογισμικού πρέπει να αντιμετωπίσει ένα πολιτιστικό ζήτημα. Οι μαθητές πρέπει να είναι εκτεθειμένοι σε βέλτιστες πρακτικές σε επίπεδο βιομηχανίας και πρέπει να μάθουν να τις αξιολογούν, προκειμένου να αποφασίσουν αργότερα ποιες θα είναι καλύτερες επιλογές για το πραγματικό περιβάλλον εργασίας τους (Coppinger, J. 2017).

Σε πρακτικό επίπεδο, οι καθορισμένες διαδικασίες διδακτικού λογισμικού μπορούν να παρέχουν ένα νήμα για πιο αποτελεσματική παροχή γνώσεων και πρακτικών σε μαθήματα. Εάν ένα μάθημα μηχανικής λογισμικού, για παράδειγμα, προσφέρει διακεκριμένα κομμάτια γνώσης, είναι δύσκολο να εκτιμηθεί το είδος της επάρκειας που αποκτήθηκε πραγματικά από τους μαθητευόμενους. Επίσης, ο χρόνος είναι

γενικά σύντομος για να εκθέσει τους σπουδαστές σε ένα σωστό σύνολο βέλτιστων πρακτικών. Ιδιαίτερα, τα προγράμματα βιομηχανικής κατάρτισης πρέπει να είναι πολύ εστιασμένα, προκειμένου να είναι αποτελεσματικά. Από την άλλη πλευρά, τα ακαδημαϊκά προγράμματα μηχανικής λογισμικού δεν είναι συνηθισμένα. Πολλοί προγραμματιστές λογισμικού προέρχονται από προγράμματα πληροφορικής, όπου η μηχανική λογισμικού είναι απλώς μια πειθαρχία μεταξύ άλλων, η οποία πρέπει να συμπιεστεί σε μερικά μαθήματα (μερικές φορές, μόνο ένα). Είναι σύνηθες να έχει η μηχανική λογισμικού ως υποχρεωτικό μάθημα που πρέπει να διδάσκεται σε ένα ή δύο τρίμηνα ή σε εξάμηνα. Συχνά, αυτό οδηγεί στην ύπαρξη τάξεων με μεγάλο αριθμό φοιτητών (Jones, Stuart Mackenzie. 2008) . Ο φόρτος εργασίας του εκπαιδευτή επιδεινώνεται από την ανάγκη συγκέντρωσης μιας μεγάλης ποικιλίας θεμάτων σε ένα συνεκτικό σύνολο. Αν ένα μάθημα λογισμικού ακολουθεί τις γραμμές μιας ευρέως σχεδιασμένης διαδικασίας ανάπτυξης λογισμικού, μπορεί να εκθέσει τους μαθητές σε ένα συνεκτικό σύνολο επιλεγμένων πρακτικών. Τα μαθήματα μπορούν να σχεδιαστούν για να παρουσιάσουν στους μαθητές ζητήματα πραγματικής ζωής και να μεταδώσουν βασικά επίπεδα επάρκειας (3D Innovations. 2013).

2.13. Κατηγορίες Εκπαιδευτικού Λογισμικού

Η εκπαιδευτική τεχνολογία και το λογισμικό είναι απαραίτητες για τη σύγχρονη τάξη. Η ταχεία υιοθέτηση ενός έξυπνου, εξορθολογισμένου λογισμικού υπήρξε ένα όφελος για τους εκπαιδευτικούς και τους διαχειριστές - και οι μαθητές το δέχθηκαν επίσης. Με λίγα λόγια, δεν υπάρχουν χαμένοι με την εκπαιδευτική τεχνολογία, γι 'αυτό δεν υπήρξαν ποτέ περισσότερες επιλογές λογισμικού. Υπάρχουν πολλοί, στην πραγματικότητα, που πιστεύουν ότι είναι δύσκολο για τους εκπαιδευτικούς και το

προσωπικό πληροφορικής τους να ξεκινήσουν να συνδυάζουν μια αποτελεσματική πλατφόρμα λογισμικού και τεχνολογίας. Ευτυχώς υπάρχουν εμπειρογνώμονες που μπορούν να παρέχουν πληροφορίες για διάφορα μέρη της αγοράς της εκπαιδευτικής τεχνολογίας (Babar, A. 2016).

Οι δάσκαλοι αναμένεται να κάνουν περισσότερα από ότι στο παρελθόν. Οι αίθουσες διδασκαλίας που συνεχίζουν να αναπτύσσονται και τα αυστηρά πρότυπα δοκιμών σημαίνουν ότι οι εκπαιδευτικοί χρειάζονται κάθε δυνατή βοήθεια. Οι σημερινοί σπουδαστές, μετά από όλα, δεν γνώρισαν ποτέ έναν κόσμο χωρίς την πανταχού παρούσα παρουσία της τεχνολογίας και καλωσορίζουν όλες τις μορφές της. Επίσης, οι σπουδαστές είναι γρήγοροι να το καταλάβουν και να αξιοποιήσουν πλήρως τα μέσα, οπότε η επένδυση σε εκπαιδευτικό λογισμικό μπορεί να έχει μακροπρόθεσμα και άμεσα οφέλη (Amidon, Jane. 2012).

Παρακάτω παρουσιάζονται ορισμένες μορφές εκπαιδευτικού λογισμικού:

Courseware - Ένα χαρτοφυλάκιο μαθημάτων και λογισμικού, το εκπαιδευτικό λογισμικό είναι, μακράν, ο πιο κοινός τύπος εκπαιδευτικού λογισμικού που διατίθεται. Το μαθήματα γινόντουσαν περίπου εφ 'όσον τα εργαστήρια ηλεκτρονικών υπολογιστών είχαν τεθεί σε χρήση για φοιτητές. Περιλαμβάνουν οποιοδήποτε λογισμικό που έχει σχεδιαστεί για να καθοδηγεί τον σπουδαστή και να μπορεί να έρθει με τη μορφή ενός εκπαιδευτικού κιτ, ή ενός εκπαιδευτικού προγράμματος για τους μαθητές. Έτσι, τόσο τα προγράμματα τέχνης των νηπιαγωγείων όσο και το συνοδευτικό λογισμικό που συνδέονται με ένα μάθημα εργαστηρίου φυσικής

θεωρούνται μαθηματικά, με αυτόν τον ορισμό. Το courseware έχει βελτιωθεί και επεκταθεί βαθιά μέσα σε λίγες μόνο δεκαετίες και ορισμένα μαθήματα διδασκαλίας μπορούν να διδάξουν ολόκληρα μαθήματα με άριστη ακρίβεια. Η ιδέα για το ποια σειρά μαθημάτων θα μπορούσε να διδαχθεί έχει επίσης αυξηθεί. Αλλά, όταν οι εταιρείες ανάπτυξης λογισμικού αναφέρονται σε μαθήματα διδασκαλίας, περιγράφουν συχνά ένα ενιαίο μάθημα (για οποιοδήποτε θέμα) που συνοδεύεται από υλικά αξιολόγησης και συμπληρωματικά μαθήματα. Ανεξάρτητα από τον ορισμό, όμως, το διδακτικό υλικό είναι η ραχοκοκαλιά του λογισμικού κάθε εκπαιδευτικού (Jones, Stuart Mackenzie. 2008).

Λογισμικό αξιολόγησης - Σε μια προσπάθεια να μειωθούν τα απόβλητα χαρτιού και οι εφιάλτες υλικοτεχνικής υποστήριξης που έρχονται με την εκτύπωση των διαγωνισμάτων, το λογισμικό αξιολόγησης έχει γίνει ένα επίκεντρο για τα σύγχρονα σχολικά συστήματα. Το λογισμικό αξιολόγησης είναι απλό, καθώς είναι σχεδιασμένο για να παρέχει δοκιμές, ή κουίζ, να καταγράφει απαντήσεις και να βαθμολογεί το αποτέλεσμα. Το λογισμικό αξιολόγησης είναι επίσης ευρέως διαδεδομένο στους εκπαιδευτικούς, καθώς είναι ένας τρόπος χαμηλής προσπάθειας για την παρακολούθηση των βαθμών σπουδαστών κατά τη διάρκεια ενός σχολικού έτους. Επειδή το λογισμικό αξιολόγησης είναι απλό, κατ 'αρχήν, υπάρχουν πολλές ανοικτές πηγές και δωρεάν επιλογές για τους εκπαιδευτικούς, οπότε δεν πρόκειται για έναν τομέα όπου απαιτούνται συνήθως μεγάλες επενδύσεις (Babar, A. 2016).

Λογισμικό αναφοράς - Το λογισμικό αναφοράς έχει προχωρήσει πλήρως στο διαδίκτυο και υπάρχουν λίγα παραδείγματα σχολικών περιοχών που εξακολουθούν να

χρησιμοποιούν αποκλειστικά λογισμικά για λόγους αναφοράς. Αυτό σηματοδοτεί μια εξαιρετικά γρήγορη αλλαγή, καθώς οι σχολικοί υπολογιστές φορτώθηκαν με λεξικό και λογισμικό εγκυκλοπαιδιών, τόσο πρόσφατα, όσο και στις αρχές της δεκαετίας του 2000. Αυτό δεν συμβαίνει πλέον, καθώς τοποθεσίες όπως η Wikipedia έχουν καταστήσει αυτό το λογισμικό παρωχημένο και εκτός από γενικούς ερευνητικούς ιστότοπους όπως το Wikipedia, το Google Scholar και το Lexis, τα περισσότερα ιατρικά και επιστημονικά περιοδικά δημοσιεύονται online. Ο ιστός προσφέρει μια πολύ μεγαλύτερη συλλογή γνώσεων από ό, τι μπορεί να ταιριάζει με ένα κομμάτι λογισμικού αναφοράς (Jones, Stuart Mackenzie. 2008).

Βοηθήματα στην τάξη - Η αγορά τεχνολογίας και λογισμικού βοήθειας στην τάξη έχει εξαντληθεί τα τελευταία χρόνια και αντιπροσωπεύει μία από τις πιο ελπιδοφόρες προσεγγίσεις στην τάξη του μέλλοντος. Τα βοηθήματα στην τάξη περιλαμβάνουν εντυπωσιακή τεχνολογία ήχου και εικόνας, συνδυάζοντας το καλύτερο οπτικοακουστικό και εκπαιδευτικό λογισμικό για να δημιουργηθεί ένα πολύ πιο συναρπαστικό σύνολο. Τα βοηθήματα στην τάξη ξεκίνησαν με φθαρμένη και ολοένα και πιο αρχαϊκή τεχνολογία, όπως ο εναέριος προβολέας και σύντομα αναπτύχθηκαν σε βελτιωμένες ψηφιακές οθόνες και τελικά σε διαδραστικούς προβολείς και λευκούς πίνακες.

2.14. Λογισμικά CAD

2.14.1. Ορισμός

Το Computer Aided Design (CAD) είναι μια τεχνολογία που αναπτύχθηκε αρχικά για να βοηθήσει τους μηχανικούς και τους αρχιτέκτονες να παράγουν μεγάλα και

σύνθετα σχέδια. Τα συστήματα CAD είναι πλέον ικανά να παράγουν εξελιγμένους εικονικούς κόσμους και χρησιμοποιούνται συνήθως εκτός της αρχικής κοινότητας-στόχου τους, σε τομείς όπως η αρχαιολογία και η βιομηχανία ψυχαγωγίας. Η επιτυχία του CAD σημαίνει ότι πολλές ζωτικές, πολύτιμες και αναντικατάστατες πληροφορίες αποθηκεύονται σε μοντέλα CAD, από τα σχέδια φορέων αεροσκαφών και ουρανοξυστών έως τα αρχεία αρχαιολογικών ανασκαφών. Σημαίνει επίσης ότι το CAD είναι ένας τομέας συνεχούς καινοτομίας και έντονου ανταγωνισμού μεταξύ πωλητών, με αποτέλεσμα συστήματα CAD που είναι εφήμερα, να είναι σε μεγάλο βαθμό ασυμβίβαστα μεταξύ τους. Είναι η αποσύνδεση μεταξύ αυτών των δύο - η αξία των μοντέλων και η εφήμερη φύση των συστημάτων - που καθιστούν την συντήρηση CAD ταυτόχρονα σημαντική και προκλητική (Amidon, Jane, 2012).

Ο Σχεδιασμός με Υπολογιστή (CAD) είναι, όπως υποδηλώνει το όνομα, η χρήση ηλεκτρονικών υπολογιστών για να βοηθήσουν στο σχεδιασμό των κατασκευασμένων προϊόντων, του δομημένου περιβάλλοντος ή των πλασματικών περιβαλλόντων. Πιο συγκεκριμένα, αναφέρεται στο λογισμικό - και αρχικά και στον ηλεκτρονικό υπολογιστή - για τη δημιουργία ψηφιακών μοντέλων φυσικών αντικειμένων. Τα συστήματα CAD είναι συνήθως δαπανηρά και περίπλοκα κομμάτια λογισμικού και οι μορφές των εγγενών αρχείων τους είναι εξίσου πολύπλοκες, αδιαφανείς και σε σημαντικό βαθμό ατελείς. Αυτό δυσχεραίνει τη διαφύλαξή τους, πράγμα που οδήγησε σε πρωτοβουλίες τυποποίησης για πάνω από 40 χρόνια και θα συνεχίσει να συμβαίνει πολύ περισσότερο στο μέλλον. Προκειμένου να διατηρηθούν τα μοντέλα CAD, είναι χρήσιμο να κατανοήσουμε το πλαίσιο στο οποίο αναπτύχθηκαν τα συστήματα αυτά, γιατί εξελίχθηκαν όπως έκαναν, πώς διαφέρουν μεταξύ τους και πώς χρησιμοποιούνται σήμερα (Babar, A. 2016).

Η άνοδος των συστημάτων CAD στη δεκαετία του 1960 οφειλόταν στην τεράστια πρακτικότητα της σχεδίασης με το χέρι. Όχι μόνο η διαδικασία ήταν επίπονη και επιρρεπή σε λάθη, αλλά θα μπορούσε να προκαλέσει και πραγματικούς πονοκεφάλους. Πολλά σχέδια σχεδιάστηκαν σε κλίμακα 1:1, κάτι που ήταν μια πρόκληση όταν έπρεπε να σχεδιαστούν φτερά αεροσκαφών, ή κύτη πλοίων. Ένας ακόμη οδηγός ήταν η ανάπτυξη το 1957 του PRONTO, του πρώτου συστήματος ψηφιακού ελέγχου ηλεκτρονικών υπολογιστών (CNC), το οποίο θα μπορούσε να χρησιμοποιηθεί για την αυτοματοποίηση ορισμένων διαδικασιών μηχανικής κατεργασίας. Ο σχεδιασμός τέτοιων συστημάτων πάνω σε χαρτί σχεδίασης ήταν, πάλι, επίπονος και επιρρεπής σε σφάλματα και θα ήταν πολύ ευκολότερο εάν τα σχήματα είχαν ήδη καθοριστεί μαθηματικά. Έτσι, από τα τέλη της δεκαετίας του 1950 έως τα μέσα της δεκαετίας του '70 υπήρξε μια έντονη προσπάθεια τόσο από τη βιομηχανία όσο και από τον ακαδημαϊκό κόσμο να βρεθούν μαθηματικές αναπαραστάσεις των σχεδίων χαρτιού και να δημιουργηθούν εργαλεία για τη συγγραφή τους. Πιθανώς το πρώτο αναγνωρίσιμο σύστημα CAD ήταν το SKETCHPAD, το οποίο αναπτύχθηκε μεταξύ 1960 και 1963 στο MIT από τον Ivan Sutherland. Η είσοδος του χρήστη πραγματοποιήθηκε μέσω ενός ελαφρού στυλό, με το οποίο ο σχεδιαστής εστίαζε στην οθόνη του υπολογιστή. Οι κύριοι φορείς της βιομηχανίας όπως οι Ford, Rénault και Lockheed ανέπτυξαν εσωτερικά συστήματα CAD στη δεκαετία του 1960 και τα πρώτα επιτυχημένα εμπορικά συστήματα CAD εμφανίστηκαν το 1969 (Amidon, Jane, 2012).

Η σημερινή έμφαση για την ανάπτυξη CAD στον τομέα της μηχανικής δεν είναι τόσο η γεωμετρική μοντελοποίηση όσο η ενσωμάτωση δεδομένων και πληροφοριών από ολόκληρο τον κύκλο ζωής του προϊόντος. Το Computer Aided Engineering (CAA) είναι ένας όρος «ομπρέλα» που συγκεντρώνει CAD, CAM (Computer Aided Manufacture), Προγραμματισμό Διαδικασιών Υπολογιστών (CAPP), Ανάλυση Πεπερασμένων Στοιχείων (FEA) και Σχεδιασμού Απαιτήσεων Υλικών (MRP) την ευρύτερη έννοια της διαχείρισης του κύκλου ζωής προϊόντων (PLM), η οποία περιλαμβάνει επίσης τομείς όπως παρακολούθηση απόδοσης, διαχείριση χαρτοφυλακίου, αρχειοθέτηση δεδομένων κ.ο.κ. (Spring. Allen, Stan. 2012). Τα συστήματα PLM διατίθενται στο εμπόριο ως ένας τρόπος ενσωμάτωσης όλων αυτών των διαφορετικών καθηκόντων. Στην αρχιτεκτονική και στις κατασκευές, η (σχεδόν) ισοδύναμη έννοια είναι η Building Information Modeling (BIM), η οποία επιδιώκει να ενσωματώσει σε ένα ενιαίο έγκυρο μοντέλο όλες τις πληροφορίες που απαιτούνται για το σχεδιασμό, την κατασκευή, τη συντήρηση και τελικά την κατεδάφιση ενός κτιρίου (Lee, Eastman, 2006). Και στις δύο περιπτώσεις, πολλοί συνεισφέροντες με διαφορετικά δικαιώματα πρόσβασης πρέπει να είναι σε θέση να συνεργάζονται για το μοντέλο, πιθανώς από διαφορετικούς ιστότοπους, έτσι ο έλεγχος της έκδοσης, ο έλεγχος πρόσβασης και ο συγχρονισμός των κύριων μοντέλων με απλοποιημένα υποκατάστατα απεικόνισης είναι κοινά χαρακτηριστικά των συστημάτων PLM και BIM. Ενώ αυτό το επίπεδο ολοκλήρωσης και ελέγχου είναι σαφώς ωφέλιμο, παρουσιάζει πρόσθετες προκλήσεις διατήρησης. Για να παραμείνουν χρήσιμα τα σχέδια ως σχέδια, θα χρειαστεί να υπάρχουν τρόποι ανάγνωσής τους όχι μόνο σε μελλοντικά συστήματα CAD, αλλά και σε όλα τα άλλα μελλοντικά συστήματα CAE. Τα αρχειακά συστήματα μπορεί να χρειαστεί να αναπαράγουν ή να παρακάμπτουν τους ελέγχους πρόσβασης των συστημάτων PLM και BIM (Babar, A. 2016).

2.14.2. Χρήσεις

Οι δυνατότητες 3D CAD modeling έχουν φανεί χρήσιμες σε πολλά περιβάλλοντα εκτός από το καθαρό προϊόν, ή τον αρχιτεκτονικό σχεδιασμό και αυτό οδήγησε στην ανάπτυξη πρόσθετων δυνατοτήτων και παροχών. Η χρήση CAD για κινούμενα σχέδια και η προσομοίωση σε πραγματικό χρόνο ώθησε την έρευνα σε πολλαπλά επίπεδα λεπτομέρειας (LOD). Αυτό φαίνεται όταν ένα μοντέλο εμπεριέχει, εκτός από την κανονική γεωμετρική παράσταση που εκφράζει με ακρίβεια τις προθέσεις του σχεδιαστή, μία, ή περισσότερες απλοποιημένες εκδοχές ενός αντικειμένου. Το τελευταίο χρησιμοποιείται όταν το αντικείμενο είναι μακρινό, μικρό, ή μερικώς συγκεχυμένο από την οπτική γωνία του θεατή, ως τρόπος επιτάχυνσης της απεικόνισης (Amidon, Jane. 2012).

Η διαδικαστική μοντελοποίηση είναι μια τεχνική όπου οι ακολουθίες των ενεργειών επεξεργασίας αποθηκεύονται ως αλγόριθμοι, ώστε να μπορούν να εφαρμοστούν αυτόματα σε 3D μοντέλα. Χρησιμοποιήθηκε αρχικά για απόδοση υψηλής ποιότητας, αλλά τώρα χρησιμοποιείται επίσης για τη δημιουργία περίπλοκης γεωμετρίας: εφαρμογή στρώματος αποχρωματισμού σε επίπεδη επιφάνεια, για παράδειγμα, ή δημιουργία ολόκληρης πόλης από μερικά δείγματα κτιρίων. Δεδομένου ότι τα αποτελέσματα τείνουν να είναι ψευδοτυχαία στη φύση - δηλαδή, προφανώς τυχαία αν και παραγόμενα θεωρητικά - η διαδικαστική μοντελοποίηση χρησιμοποιείται πιο εκτεταμένα στη δημιουργία εικονικών κόσμων και όχι σε επίσημο σχεδιαστικό έργο. Για περίπου όσο υπάρχει ενδιαφέρον για τη δημιουργία ψηφιακών αναπαραστάσεων φανταστικών αντικειμένων, υπάρχει επίσης ενδιαφέρον για τη δημιουργία ψηφιακών μοντέλων των υπαρχόντων αντικειμένων. Ορισμένα συστήματα CAD μπορούν να

λάβουν δεδομένα από μηχανές μέτρησης συντεταγμένων (CMM), ή σαρωτές υπολογιστικής τομογραφίας (CT) και να δημιουργήσουν μοντέλα CAD. Στη βιομηχανία τέτοια μοντέλα θα μπορούσαν να χρησιμοποιηθούν για να συγκρίνουν τα κατασκευαζόμενα εξαρτήματα με τα αρχικά τους σχέδια για να αναλύσουν τη φθορά, ή ως αρχικό βήμα προς την αντίστροφη μηχανουργική κατασκευή ενός αντικειμένου. Είναι επίσης δυνατό, φυσικά, να εισαγάγουμε ένα σύνολο σημείων δεδομένων σε ένα σύστημα CAD και να «συνδέσουμε χειροκίνητα τις κουκίδες» για να δημιουργήσουμε ένα μοντέλο CAD. Πρόκειται για μια τυπική περίπτωση χρήσης στην αρχαιολογία, όπου το CAD έχει βρει ευρεία χρήση ως εργαλείο τεκμηρίωσης τόσο για τα σχέδια 2D τοποθεσιών όσο και για τα τρισδιάστατα μοντέλα ιστοτόπων και αντικειμένων (Jones, Stuart Mackenzie, 2008) .

Ένα από τα αξιοθέατα του λογισμικού CAD τόσο για τους αρχαιολόγους όσο και για τους αρχιτέκτονες είναι η ικανότητά του να οργανώνει δεδομένα σε στρώματα τα οποία μπορούν να προβληθούν ή να κρυφτούν ανάλογα με τις ανάγκες. Η χρήση των επιπέδων είναι έτσι ενσωματωμένη στην αρχιτεκτονική ροή εργασίας, η οποία περιγράφει ποιες πληροφορίες πρέπει να συμπεριληφθούν σε κάθε μία και πώς να ονομάζονται, αποτελούν αντικείμενο εθνικών και διεθνών προτύπων (π.χ., το εθνικό πρότυπο CAD των ΗΠΑ, ISO 13567). Στην αρχαιολογία, τα στρώματα χρησιμοποιούνται με πολύ πιο ποικίλο τρόπο ανάλογα με την ανάγκη: να διαχωρίζονται διαφορετικά υλικά, στρώματα, φάσεις κατασκευής, λειτουργίες κλπ. για να διαχωριστούν τα δεδομένα χάρτη από άδεια από πρόσφατα συλλεχθέντα δεδομένα, για παράδειγμα. και τον διαχωρισμό των φυσικών χαρακτηριστικών από τους σχολιασμούς. Επομένως, κατά τη διατήρηση μοντέλων CAD, είναι σημαντικό οι καταγραφές των επιπέδων να είναι επαρκώς τεκμηριωμένες, ώστε να είναι γνωστή η

σημασία των δεδομένων. Τέτοιες συμβάσεις πρέπει να αποτελούν μέρος οποιουδήποτε σχεδίου διαχείρισης αρχαιολογικών δεδομένων και η τεκμηρίωσή τους να μεταφέρεται στο αρχείο όταν κατατίθεται το μοντέλο CAD.

Δεδομένου ότι η αρχαιολογία είναι μια καταστρεπτική επιστήμη, εφόσον η κάθε περιοχή έχει σκαφθεί, τα αρχεία που παράγονται από την ομάδα ανασκαφής γίνονται η μόνη πηγή πληροφοριών σχετικά με την εκάστοτε τοποθεσία: οι πληροφορίες δεν μπορούν να αναδημιουργηθούν αργότερα. Είναι επομένως πολύ σημαντικό όταν οι αρχαιολόγοι δεσμεύουν τέτοιες πληροφορίες σε ένα μοντέλο CAD, να έχουν την πεποίθηση ότι το μοντέλο θα είναι ακόμα ευανάγνωστο και επαναχρησιμοποιούμενο για πολύ καιρό στο μέλλον. Επίσης, σε μηχανικά και αρχιτεκτονικά πλαίσια, υπάρχουν ισχυροί κινητήριοι παράγοντες για τη διασφάλιση της μακροπρόθεσμης χρήσης των σχεδίων, όπως επίσης και των νομικών και κανονιστικών απαιτήσεων για αυτό. Αυτές οι απαιτήσεις οφείλονται στην ίδια την ανάγκη να γνωρίζουμε, εαν τα πράγματα πάνε στραβά, γιατί ενδέχεται να πάνε στραβά και πώς αυτά τα περιστατικά μπορούν να αποφευχθούν στο μέλλον. Σε τέτοιες έρευνες ο σκοπός των σχεδιαστών μπορεί να είναι εξίσου σημαντικός με την εξέταση του προϊόντος όπως κατασκευάζεται ή κατασκευάζεται και γι 'αυτό θα το κάνουν μόνο τα αρχικά σχέδια και τα δικαιολογητικά. Οι απαιτήσεις του πελάτη είναι πιθανό να αλλάξουν αρκετές φορές κατά τη διάρκεια ζωής ενός αεροσκάφους, ή ενός νοσοκομείου, δηλαδή, που σημαίνει ότι θα πρέπει να γίνουν προσαρμογές και τροποποιήσεις. Αυτό θα είναι πολύ πιο εύκολο αν ο σχεδιαστής, ή ο αρχιτέκτονας έχει πρόσβαση στο ιστορικό των κατασκευών και τη σημασιολογία του αρχικού σχεδίου και ίσως απαγορευτικά δαπανηρή εάν ο σχεδιασμός πρέπει να ανασχηματιστεί, ή να αναδημιουργηθεί εκ νέου (Freecadweb.org. n.d.).

2.14.3. Τύποι

Σε γενικές γραμμές, ένα πακέτο CAD (Computer Aided Design) έχει τρία συστατικά: α) Σχεδίαση, β) Ανάλυση, και γ) Οπτικοποίηση,. Ακολουθεί μια σύντομη περιγραφή αυτών των στοιχείων (Jones, Stuart Mackenzie. 2008).

α) Σχεδιασμός: Ο σχεδιασμός αναφέρεται στη γεωμετρική μοντελοποίηση, δηλ. 2-D και 3-D μοντελοποίηση, συμπεριλαμβανομένης της σύνταξης, της δημιουργίας μερών, της δημιουργίας σχεδίων με διάφορες όψεις του τμήματος, των συγκροτημάτων των τμημάτων κλπ.

β) Ανάλυση: Η ανάλυση αναφέρεται σε αναλύσεις πεπερασμένων στοιχείων, βελτιστοποίηση και άλλες αναλύσεις μηχανικής. Γενικά, δημιουργείται πρώτα ένα γεωμετρικό μοντέλο και στη συνέχεια το μοντέλο αναλύεται για φορτία, τάσεις, στιγμιαία αδράνεια και όγκο κλπ.

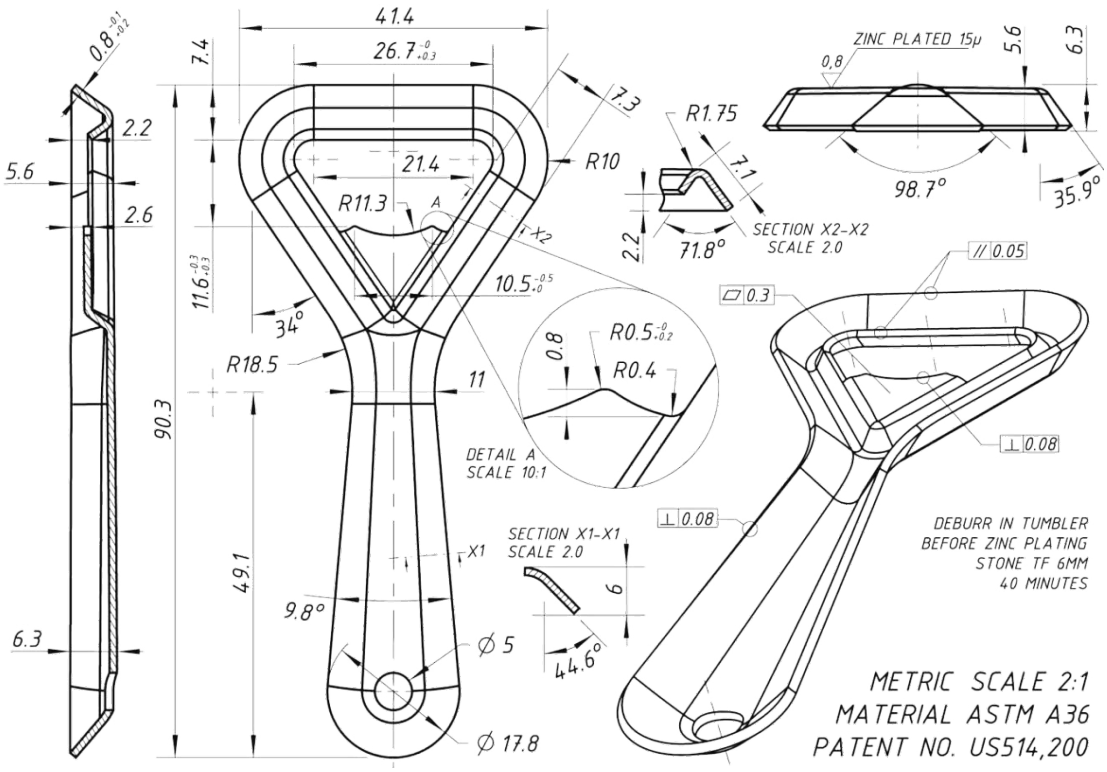
γ) Οπτικοποίηση: Η απεικόνιση αναφέρεται στα γραφικά των υπολογιστών, τα οποία περιλαμβάνουν: την απόδοση ενός μοντέλου, τη δημιουργία σχεδιαγραμμάτων μορφής πίτας, περιγράμματος, σκίαση ενός μοντέλου, μεγέθους, κινούμενων σχεδίων κ.λπ.

Κάθε μία από αυτές τις τρεις κατηγορίες έχει αναπτυχθεί εκτενώς τα τελευταία 30 χρόνια. Αρκετά βιβλία είναι γραμμένα για κάθε ένα από αυτά και τα σχετικά μαθήματα είναι διαθέσιμα μέσω των ακαδημαϊκών ιδρυμάτων και της βιομηχανίας.

Τα περισσότερα εμπορικά πακέτα CAD (λογισμικό) αποτελούνται από ένα μόνο συστατικό: σχεδιασμό-ανάλυση-οπτικοποίηση. Ωστόσο μερικοί από τους προμηθευτές έχουν αναπτύξει ένα ολοκληρωμένο πακέτο που περιλαμβάνει όχι μόνο

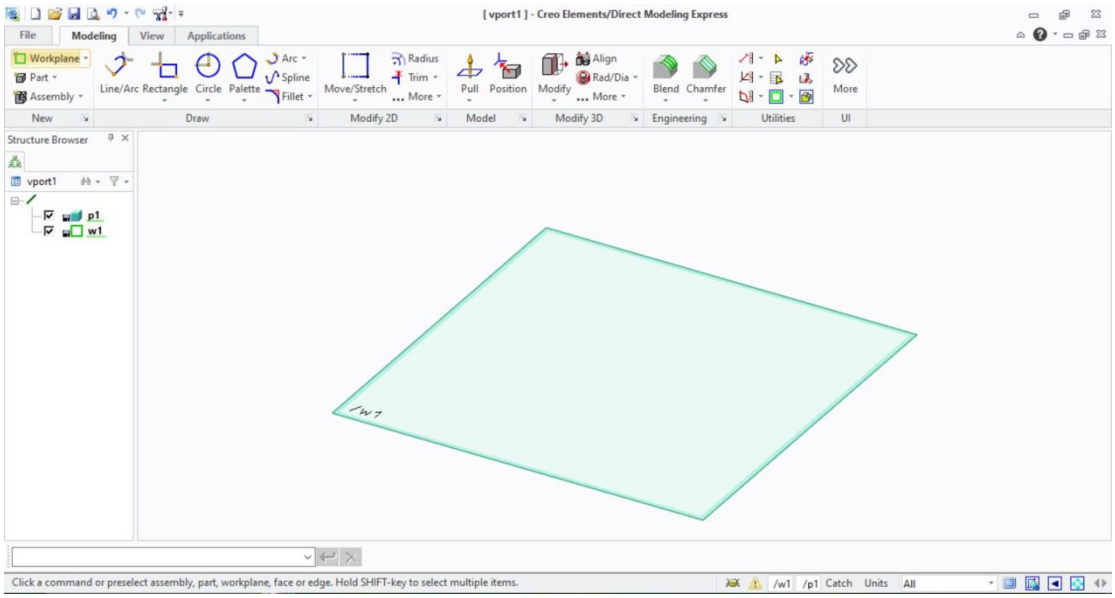
αυτούς τους τρεις τομείς, αλλά περιλαμβάνει επίσης το λογισμικό παραγωγής (CAM). Λόγω της μεγάλης απαίτησης αποθήκευσης, τα ολοκληρωμένα πακέτα χρησιμοποιούν είτε σταθμό εργασίας UNIX είτε πλατφόρμα mainframe και όχι τη δημοφιλή πλατφόρμα PC. Με τη βελτίωση της ταχύτητας υπολογιστών υπολογιστών, είναι μόνο θέμα χρόνου πριν δούμε ένα ολοκληρωμένο πακέτο να τρέχει σε έναν υπολογιστή. Το CAD έχει φέρει επανάσταση στη σύγχρονη μηχανική πρακτική. μικρές και μεγάλες εταιρείες το χρησιμοποιούν με τον ίδιο τρόπο, ξοδεύοντας αρκετά δισεκατομμύρια δολάρια για την αρχική αγορά ή μίσθωση μόνη της. Οι θέσεις εργασίας που σχετίζονται με το CAD έχουν μεγάλη ζήτηση και οι νέοι πτυχιούχοι έχουν πλεονεκτήματα έναντι των ανώτερων συναδέλφων τους, καθώς είναι πιο ενημερωμένοι και πιο παραγωγικοί (Amidon, Jane, 2012).

Παρακάτω παρουσιάζονται ορισμένα γνωστά ελεύθερα CAD λογισμικά. Τα λογισμικά αυτά θα συγκριθούν με το Autocad για σχεδιασμό σε 2 διαστάσεις και με το Creo Parametric για τρισδιάστατη σχεδίαση. Το σχέδιο είναι το ακόλουθο:



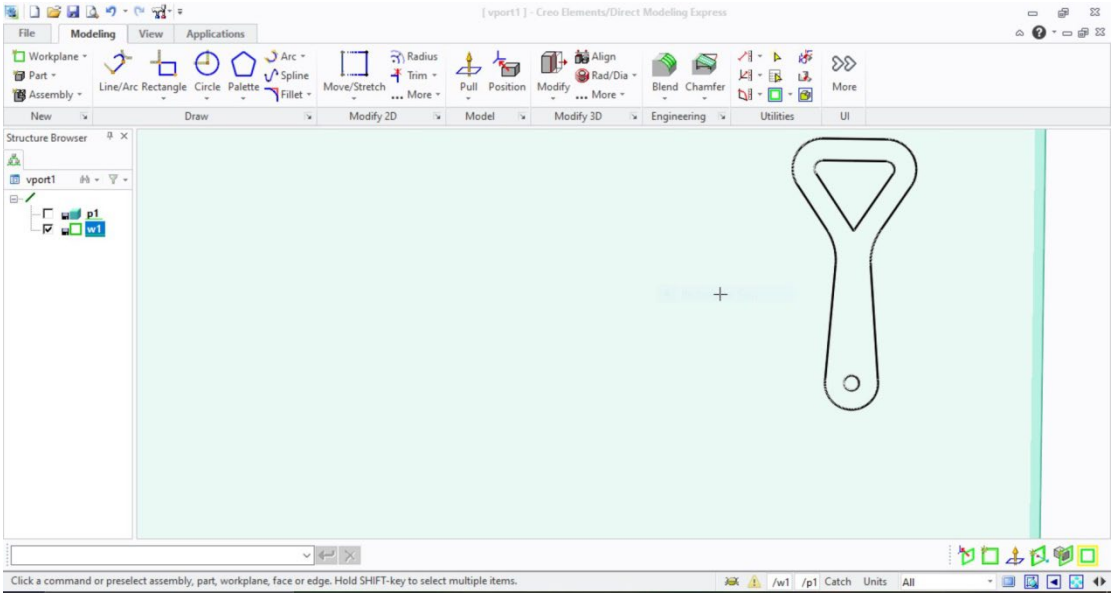
2.14.3.1. OCUS Benchmark v. 7

Creo Elements/Direct: Εύκολο στη χρήση και το πιο ισχυρό διαθέσιμο δωρεάν πρόγραμμα μοντελοποίησης 3D CAD. Δεν χρησιμοποιεί ιστορικό μοντελοποίησης, ανταυτού δίνεται η δυνατότητα απευθείας κλικ σε οποιοδήποτε σημείο του μοντέλου για να προστεθούν, ή να ρυθμιστούν χαρακτηριστικά. Είναι μια ελαφρύτερη έκδοση του εμπορικού προϊόντος τους, ώστε τα συγκροτήματα περιορίζονται σε 60 μέρη.



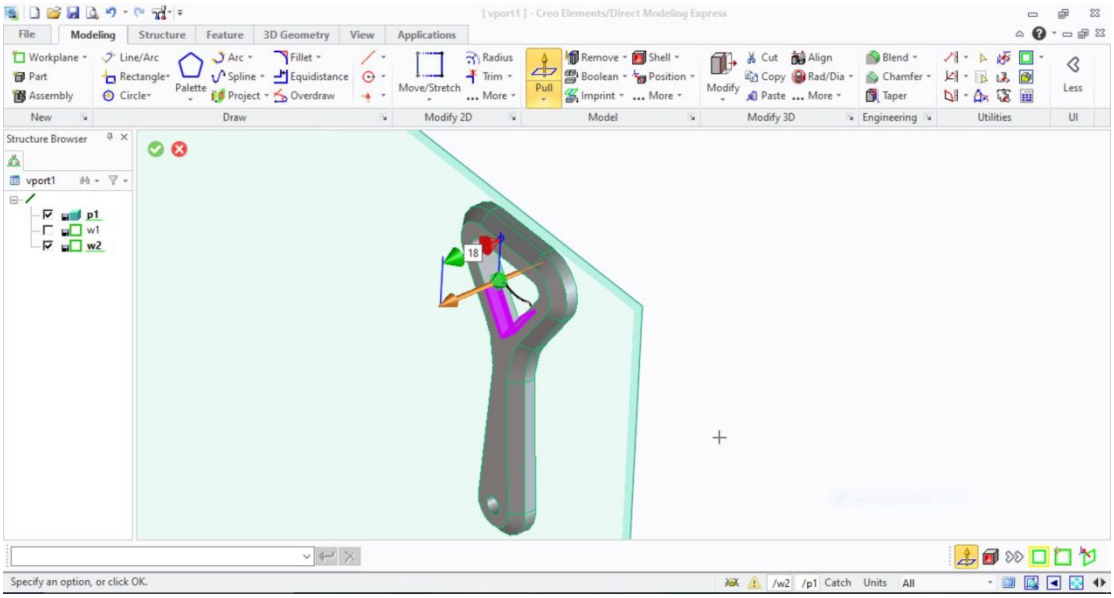
2.14.3.2. Διεπαφή του Creo Direct/Elements

Το γεγονός ότι σε αντίθεση με το εμπορικό προϊόν δεν χρησιμοποιεί ιστορικό μοντελοποίησης το καθιστά πρακτικά δύσχρηστο, εφόσον ότι επεξεργασία γίνεται στο αντικείμενο σχεδιασμού, είτε σε 2 διαστάσεις, είτε τρισδιάστατα το πρόγραμμα εντάσσει οποιαδήποτε τροποποίηση στο κομμάτι (part) και το μόνο που έχει πέραν αυτού ως αναφορά είναι το επίπεδο σχεδίασης (workplane). Η έλλειψη συγκρατήσεων (constraints) είναι επίσης αισθητή και μπορεί να επηρεάσει το σχεδιασμό, καθιστώντας κάποιες εντολές ποιο δύσκολες στην χρήση, λόγω μορφολογίας ενός αντικειμένου και των εντολών που επιλέγονται κατά την σχεδίαση και μορφοποίηση του.

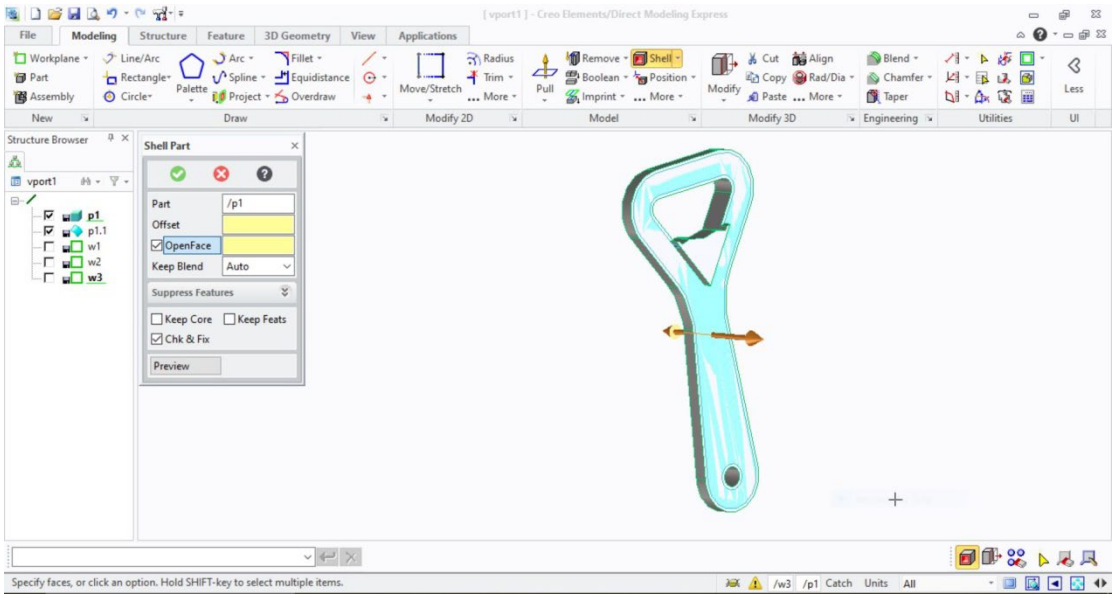


2.14.3.3. Διδυάστατη Σχεδίαση στο Creo Elements/Direct

Το λογισμικό δεν έχει το ιστορικό σχεδίασης που διαθέτει το εμπορικό προϊόν με αποτέλεσμα οι εντολές σχεδιασμού να «στοιβάζονται» κάτω από το ίδιο παραμετρικό στερεό, ή σχέδιο 2 διαστάσεων δουλεύει ο χρήστης κάθε φορά. Το μόνο άλλο σημείο αναφοράς του χρήστη είναι το επίπεδο σχεδίασης (workplane) που δουλεύει:

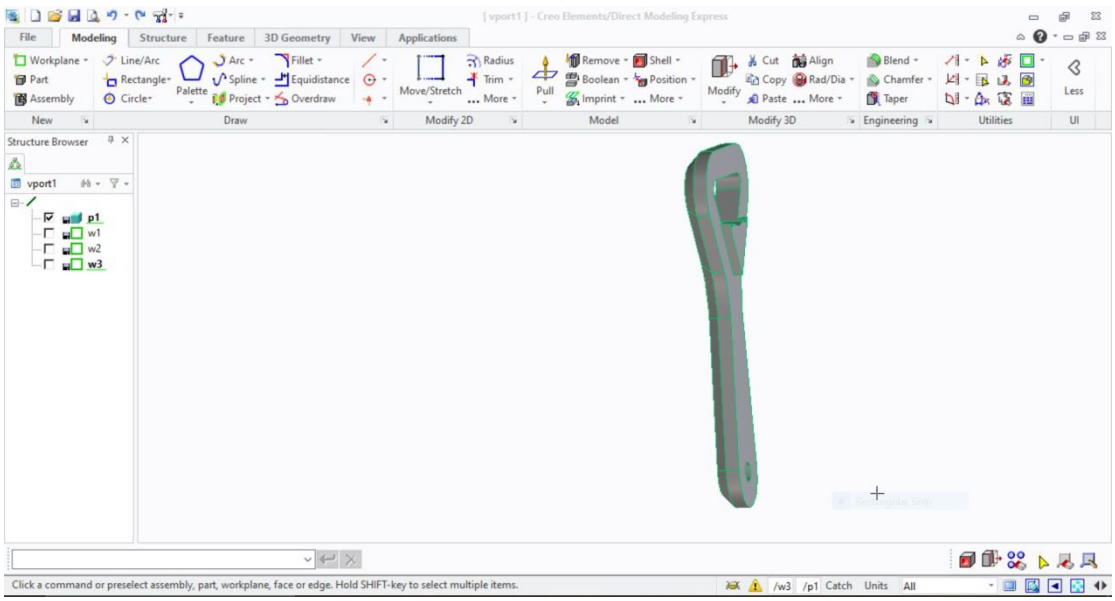


2.14.3. Εντολή εξώθησης (pull)



2.14.3.4. Εντολή κουφώματος (shell)

Συνεπώς, η έλλειψη ιστορικού σχεδίασης σημαίνει ότι το σχέδιο δεν επιδέχεται τροποποιήσεων (edit) σε προηγούμενες εντολές - πέραν της άμεσης εντολής αναίρεσης μιας διαδικασίας (undo) - σε περίπτωση που χρειαστεί να αλλάξει κάποιο στοιχείο, ή διάσταση του σχεδίου, διότι όλα εντάσσονται στο κομμάτι σχεδίασης (part).



2.14.3.5. Λάθος χρήση της εντολής εξώθησης, οδηγεί σε παραποίηση του τελικού σχεδίου

Solid Edge Student Edition: Ένα δωρεάν 3D παραμετρικό πρόγραμμα CAD. Αρχικώς δημιουργήθηκε κι αναπτύχθηκε από την εταιρεία Intergraph χρησιμοποιώντας το στερεό πρότυπο γεωμετρικής μοντελοποίησης ACIS, αλλά στην συνέχεια άλλαξε στο πρότυπο Parasolid, όταν αγοράστηκε από την USG το 1998. Το 2007 η USG αγοράστηκε από την Siemens και το Solid Edge πέρασε στον όμιλο Siemens. Εκτός από την τριδιάστατατη εκδοχή τους, η Siemens έβγαλε το 2006 μια διδιάστατη freeware εκδοχή, ονόματι Solid Edge 2D Drafting.

Η διαδικασίες μοντελοποίησης του λογισμικού είναι 3:

- **Διατεταγμένη Μοντελοποίηση**

Η διαδικασία διατεταγμένης μοντελοποίησης ξεκινά με ένα βασικό χαρακτηριστικό που ελέγχεται από ένα δισδιάστατο σκίτσο, το οποίο είναι είτε γραμμικό, περιστρεφόμενο, υπερυψωμένο ή σαρωμένο. Κάθε επόμενο χαρακτηριστικό βασίζεται στο προηγούμενο χαρακτηριστικό. Κατά την επεξεργασία, το μοντέλο "επαναφέρεται" στο σημείο όπου δημιουργήθηκε το χαρακτηριστικό, έτσι ώστε ο χρήστης να μην μπορεί να προσπαθήσει να εφαρμόσει περιορισμούς σε γεωμετρία που δεν υπάρχει ακόμη. Το μειονέκτημα είναι ότι ο χρήστης δεν βλέπει πώς η επεξεργασία θα αλληλεπιδράσει με τα επόμενα χαρακτηριστικά. Αυτό συνήθως ονομάζεται μοντελοποίηση βάσει ιστορικού, ή αναγέννησης. Τόσο στη διατεταγμένη όσο και στη σύγχρονη λειτουργία, το Solid Edge προσφέρει ισχυρή, εύκολη, αλλά σταθερή μοντελοποίηση σε υβριδική λειτουργία επιφάνειας/στερεάς, όπου η

τεχνολογία Rapid Blue βοηθά στη δημιουργία σύνθετων σχημάτων διαισθητικά και εύκολα.

- **Άμεση Μοντελοποίηση**

Οι λειτουργίες Άμεσης Μοντελοποίησης επιτρέπουν την αλλαγή της γεωμετρίας και της τοπολογίας του μοντέλου χωρίς να παρεμποδίζεται από τα υπάρχοντα, ή την έλλειψη παραμετρικών δεδομένων, ή δεδομένων ιστορικού ενός εγγενούς μοντέλου σε ένα εισαγόμενο μοντέλο. Αυτό είναι πολύ χρήσιμο για την εργασία με εισαγόμενα μοντέλα, ή σύνθετα εγγενή μοντέλα. Οι λειτουργίες Άμεσης Μοντελοποίησης είναι διαθέσιμες τόσο σε Ταξινομημένη, όσο και σε Σύγχρονη λειτουργία. Εάν χρησιμοποιηθούν σε Ταξινομημένη λειτουργία, οι άμεσες επεξεργασίες μοντελοποίησης προσαρτώνται στο δέντρο ιστορικού στο σημείο της τρέχουσας επαναφοράς, όπως ακριβώς και κάθε άλλο ταξινομημένο χαρακτηριστικό.

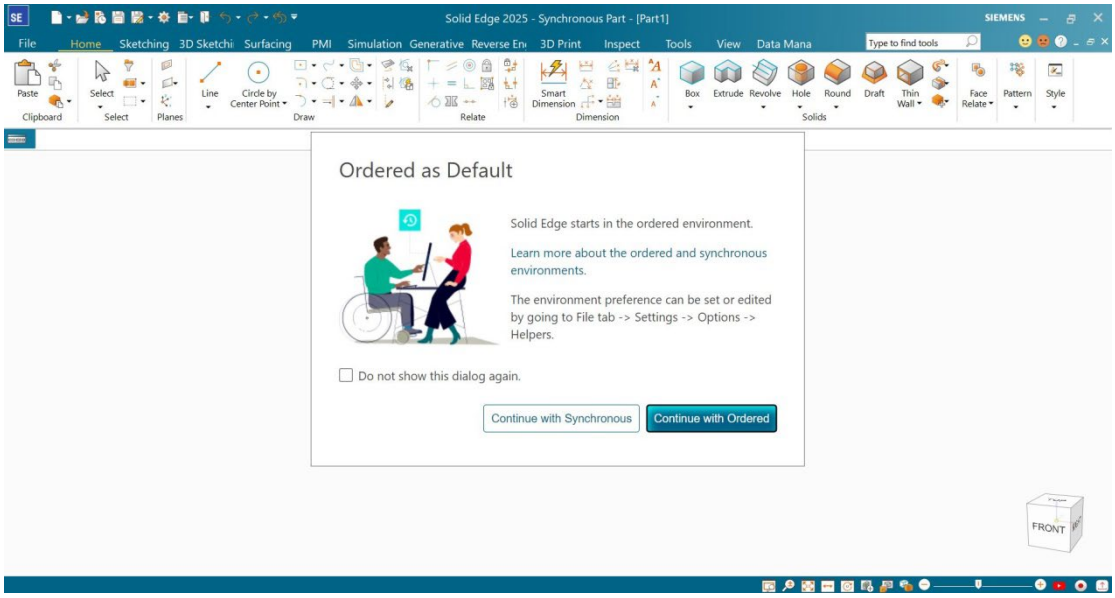
- **Σύγχρονη Μοντελοποίηση**

Το λογισμικό συνδυάζει την άμεση μοντελοποίηση με τον σχεδιασμό που βασίζεται στις διαστάσεις (χαρακτηριστικά και σύγχρονη επίλυση παραμετρικών) με την ονομασία "Σύγχρονη Τεχνολογία". Οι παραμετρικές σχέσεις μπορούν να εφαρμοστούν απευθείας στα χαρακτηριστικά του στερεού χωρίς να χρειάζεται να εξαρτώνται από τη γεωμετρία του διδιάστατου σκίτσου και οι κοινές παραμετρικές σχέσεις εφαρμόζονται αυτόματα.

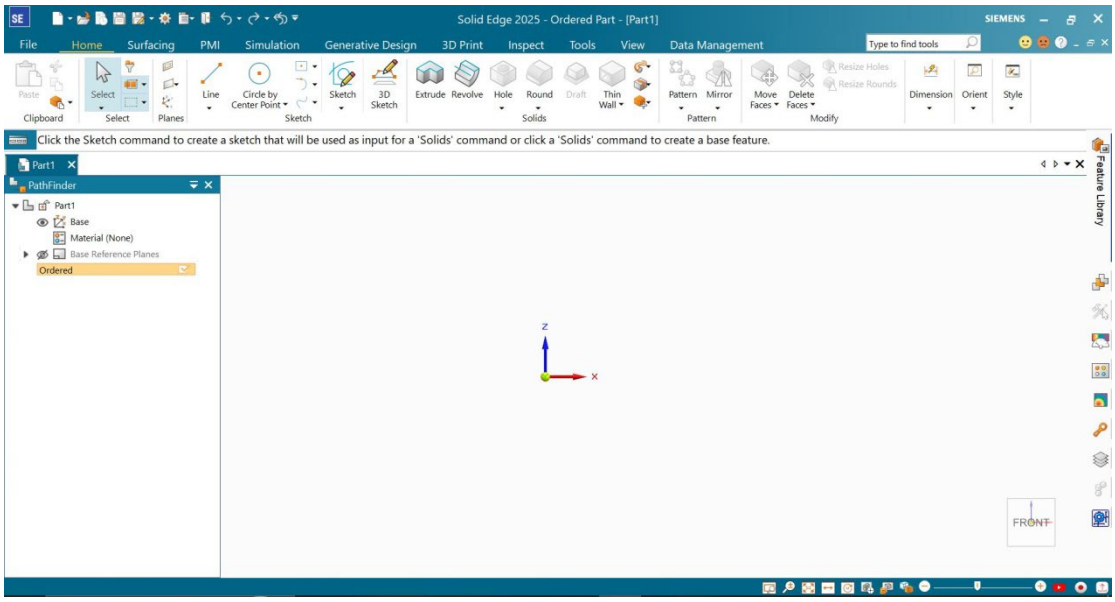
Σε αντίθεση με άλλα συστήματα άμεσης μοντελοποίησης δεν καθοδηγείται από το τυπικό σύστημα μοντελοποίησης που βασίζεται στο ιστορικό, αλλά παρέχει παραμετρική μοντελοποίηση που βασίζεται στις διαστάσεις συγχρονίζοντας τη γεωμετρία, τις παραμέτρους και τους κανόνες χρησιμοποιώντας μια μηχανή λήψης αποφάσεων, επιτρέποντας στους χρήστες να εφαρμόζουν απρόβλεπτες αλλαγές. Αυτό το μοντέλο επεξεργασίας που βασίζεται σε αντικείμενα είναι γνωστό ως Διεπαφή Δράσης Αντικειμένου, η οποία δίνει έμφαση σε μια Διεπαφή Χρήστη που παρέχει Άμεση Χειραγώγηση αντικειμένων (DMUI). Το ST2 πρόσθεσε υποστήριξη για σχεδιασμό λαμαρίνας, καθώς και για την αναγνώριση καμπυλώσεων, πτυχώσεων και άλλων χαρακτηριστικών εισαγόμενων εξαρτημάτων λαμαρίνας.

Η Σύγχρονη Τεχνολογία έχει ενσωματωθεί στο Solid Edge και σε ένα άλλο εμπορικό λογισμικό CAD της Siemens, το NX, ως επίπεδο εφαρμογής που βασίζεται στα στοιχεία λογισμικού D-Cubed και Parasolid.

Η λειτουργία και ο τρόπος χρήσης των εντολών σχεδίασης είναι αρκετά παρόμοια με αυτών του Creo Parametric, είτε σε σχεδιασμό 2 διαστάσεων, είτε για τριδιάστατη σχεδίαση. Το πρόγραμμα ρωτάει τον χρήστη ποιο είδους μοντελοποίησης επιλέγει να χρησιμποποιήσει, μόλις ξεκινήσει κάποιο νέο κομμάτι σχεδίασης:



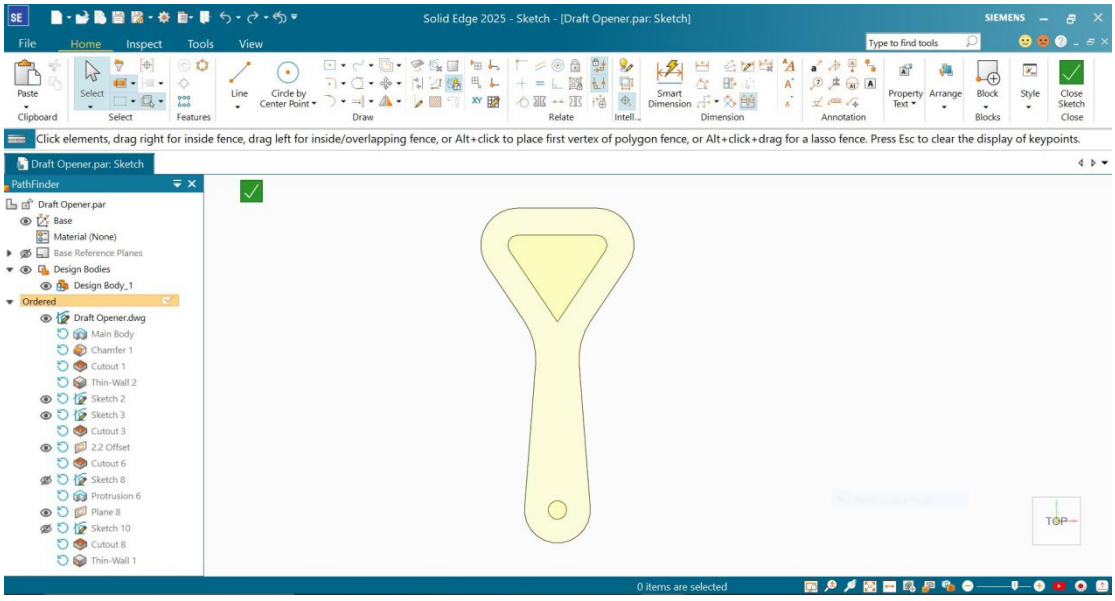
2.14.3.6. Επιλογή Σύγχρονης, ή Διατεταγμένης Μοντελοποίσης



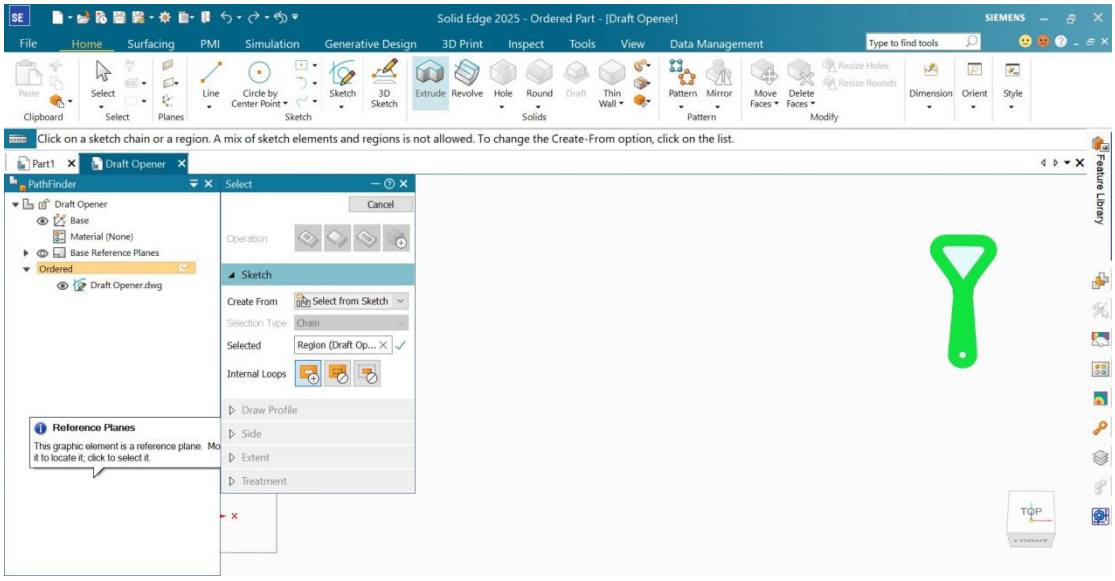
2.14.3.7. Διεπαφή του Solid Edge Διατεταγμένης Μοντελοποίσης με το ιστορικό σχεδίασης στα αριστερά

Οι εντολές της τριδιαάστατης μοντελοποίησης ακολουθούν το ίδιο μοτίβο, με την διαφορά ότι λόγω της Διατεταγμένης μοντελοποίησης οι εντολές βασίζονται πάνω στο διδιάστατο σκίτσο, δίχως να υπάρχουν συγκρατήσεις (constraints) με αποτέλεσμα το σχέδιο να μπορεί να αλλάξει ανά πάσα στιγμή, το οποίο είναι και καλό και κακό αναλόγως των τροποποιήσεων και της αναδημιουργίας του σχήματος:

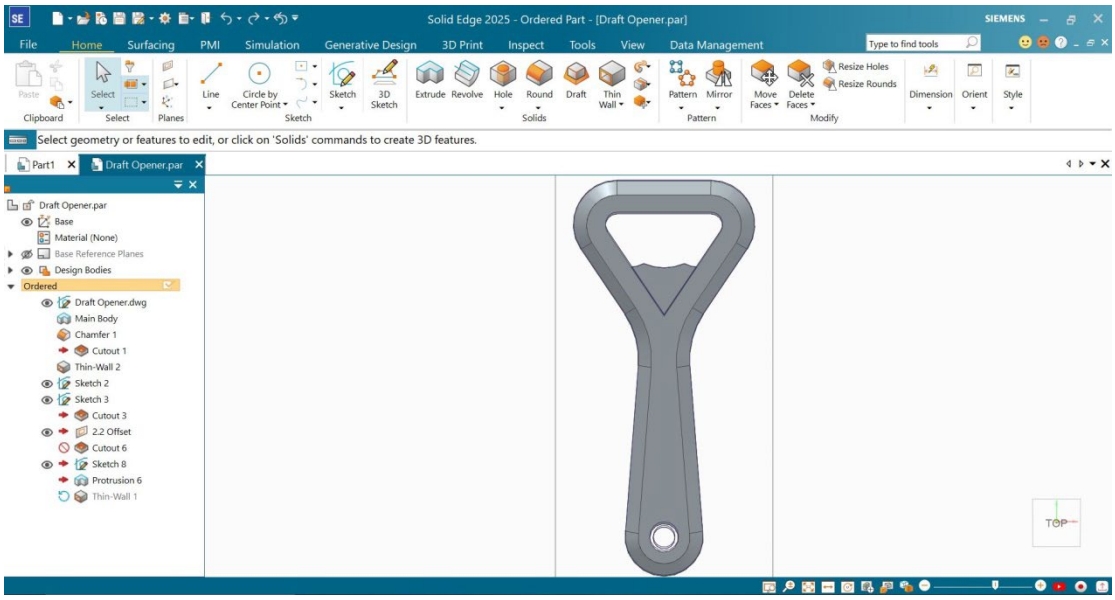
Από την διεπαφή μέχρι την γραμμή εντολών και τις κορδέλες σχεδίασης και όψεων, εώς και τα επίπεδα σχεδίασης, το Solid Edge δεν απέχει πολύ από το Creo όσον αφορά τη διαδιακασία του σχεδίου. Η επιλογή μοντελοποίησης που χρησιμοποιήθηκε ήταν διατεταγμένη μοντελοποίηση, εξαιτίας της πρακτικότητας του «αναδημιουργούμενου» μοντέλου κατά την διαδικασία της σχεδίασης, τόσο σε 2 διαστάσεις, όσο και στην τριδιάστατη σχεδίαση. Αυτό παρέχει την δυνατότητα για τροποποίηση προηγούμενων διαδικασιών και διαρκή επιστροφή στη βάση του σχεδίου, που είναι το διδιάστατο σχέδιο:



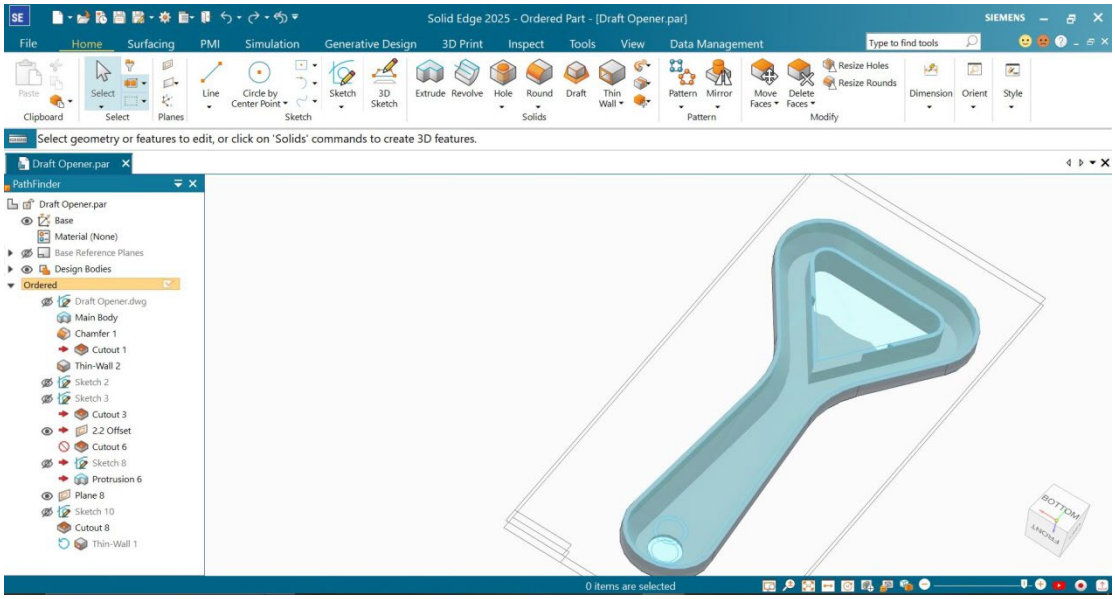
2.14.3.8. Δισδιάστατη απεικόνιση του σχεδίου



2.14.3.9. Εντολή Εξώθησης (Extrude)

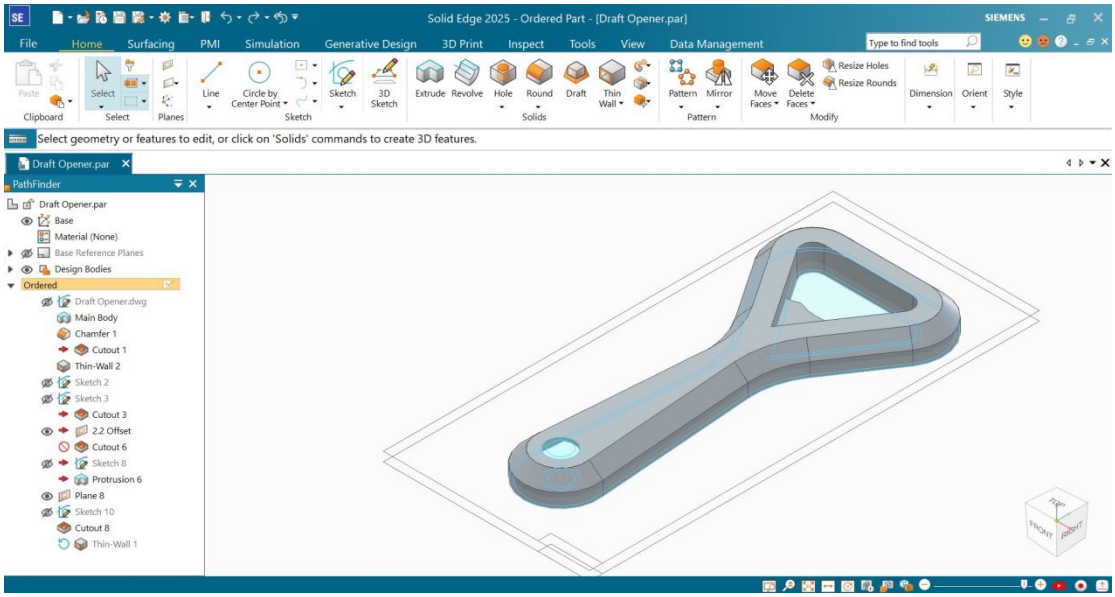


2.14.3.10. Μπροστινή όψη του σχεδίου



2.14.3.11. Πίσω όψη του σχεδίου

Η μόνη διαφορετική εντολή για την εκτέλεση αυτού του σχεδίου, είναι η εντολή δημιουργίας κουφώματος, η οποία εδώ ονομάζεται Thin Wall. Το τελικό σχέδιο φαίνεται στην παρακάτω απεικόνιση:

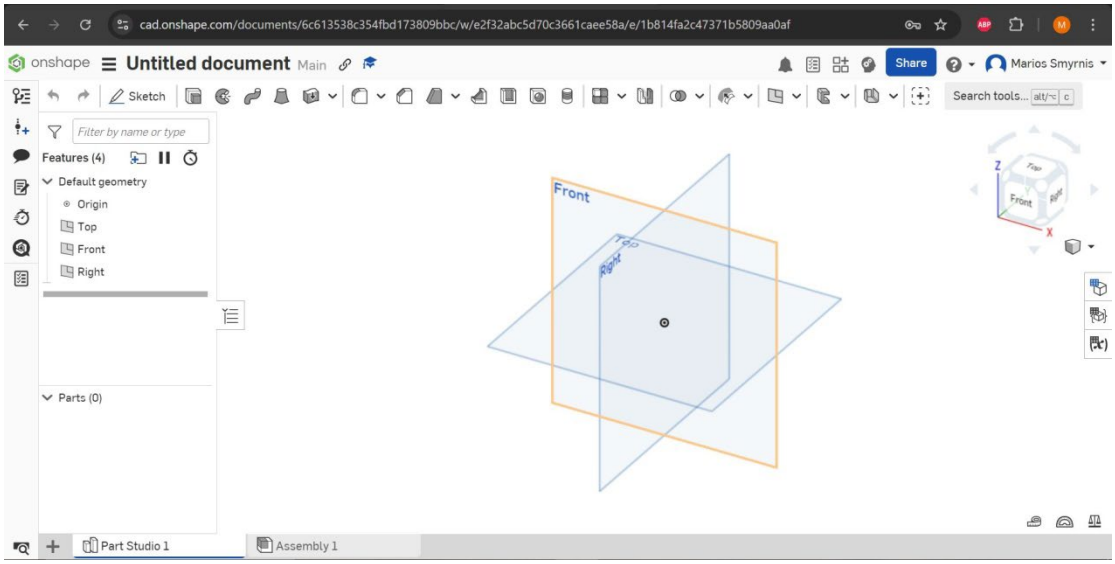


2.14.3.12. Ισομετρική όψη απεικόνισης σχεδίου

PTC Oshape: Το Onshape είναι ένα σύστημα λογισμικού σχεδιασμού με τη βοήθεια υπολογιστή (CAD), το οποίο παρέχεται μέσω Διαδικτύου μέσω ενός μοντέλου

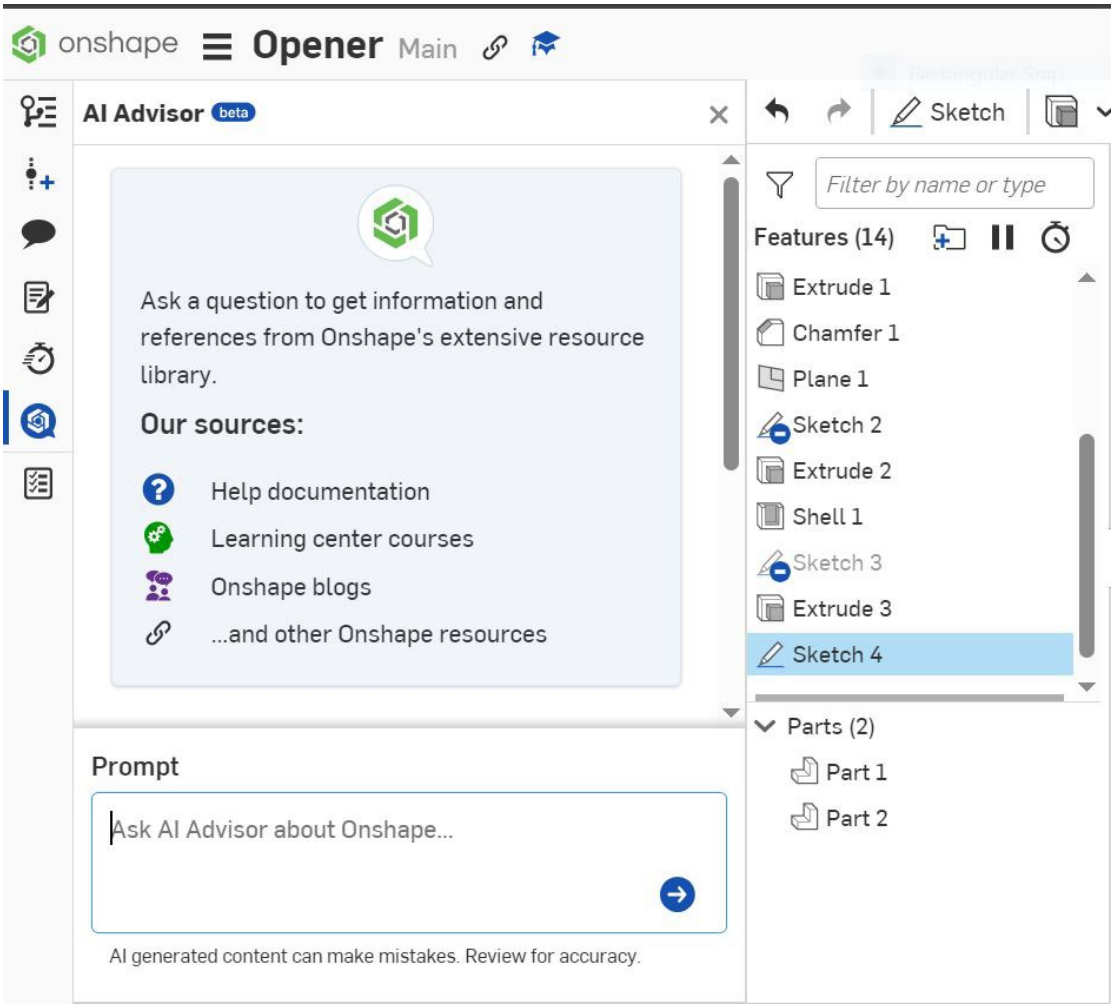
λογισμικού ως υπηρεσίας (SaaS). Κάνει εκτεταμένη χρήση του cloud computing, με επεξεργασία και απόδοση που απαιτούν μεγάλη υπολογιστική ισχύ και εκτελούνται σε διακομιστές που βασίζονται στο Διαδίκτυο, και οι χρήστες μπορούν να αλληλεπιδρούν με το σύστημα μέσω ενός προγράμματος περιήγησης ιστού ή των εφαρμογών iOS και Android. Ως σύστημα SaaS, οι αναβαθμίσεις του Onshape κυκλοφορούν απευθείας στη διεπαφή ιστού και το λογισμικό δεν απαιτεί συντήρηση από τον χρήστη.

Το Onshape επιτρέπει στις ομάδες να συνεργάζονται σε ένα ενιαίο κοινόχρηστο σχέδιο, με τον ίδιο τρόπο που πολλοί συγγραφείς μπορούν να συνεργαστούν για την επεξεργασία ενός κοινόχρηστου εγγράφου μέσω υπηρεσιών cloud. Εστιάζεται κυρίως στο μηχανολογικό CAD (MCAD) και χρησιμοποιείται για σχεδιασμό προϊόντων και μηχανημάτων σε πολλούς κλάδους, συμπεριλαμβανομένων των καταναλωτικών ηλεκτρονικών ειδών, των μηχανολογικών μηχανημάτων, των ιατρικών συσκευών, της τρισδιάστατης εκτύπωσης, των εξαρτημάτων μηχανών και του βιομηχανικού εξοπλισμού. Επειδή είναι σχεδιασμένο από την ίδια εταιρεία όπως το βιομηχανικό προϊόν τους, η διεπαφή του προγράμματος και οι εντολές σχεδιασμού είναι ίδιες:



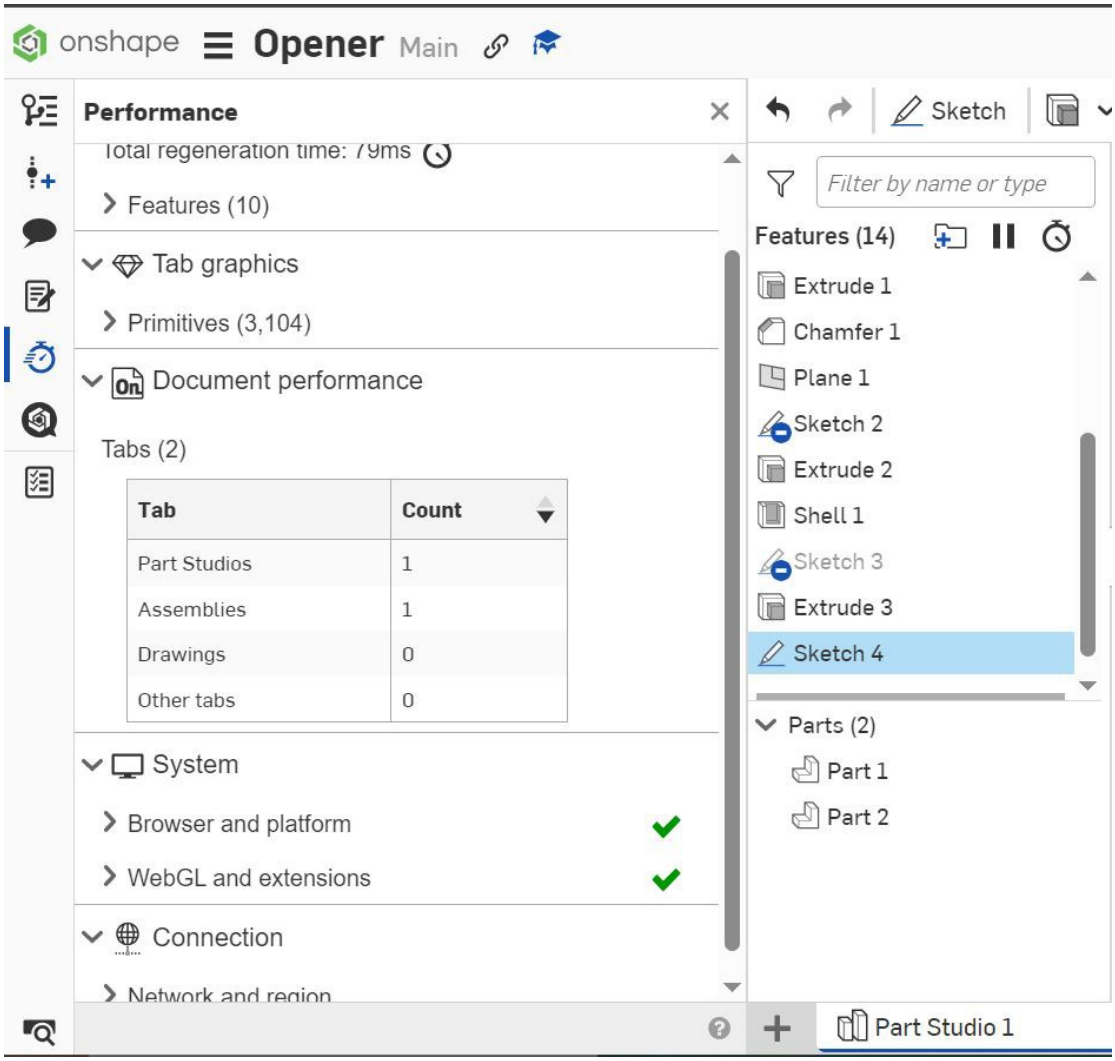
2.14.3.13. Διεπαφή Onshape

Λόγω της φύσης του ως Cloud Sharing πρόγραμμα το Onshape δεν έχει επιλογή αποθήκευσης της πορείας εργασίας του χρήστη επάνω σε κάποιο σχέδιο, ωστόσο εφόσον υπάρχει πρόσβαση στο διαδίκτυο, η δουλειά που έχει γίνει πάνω σε ένα, ή περισσότερα σχέδια παραμένει αναλείωτη από την τελευταία εντολή που έδωσε ο χρήστης, μέχρι να αποσυνδεθεί από την πλατφόρμα. Το λογισμικό απαιτεί πρόσβαση στο διαδίκτυο και παρέχει δυνατότητα βοήθειας από τεχνητή νοημοσήνη (AI Assistant):



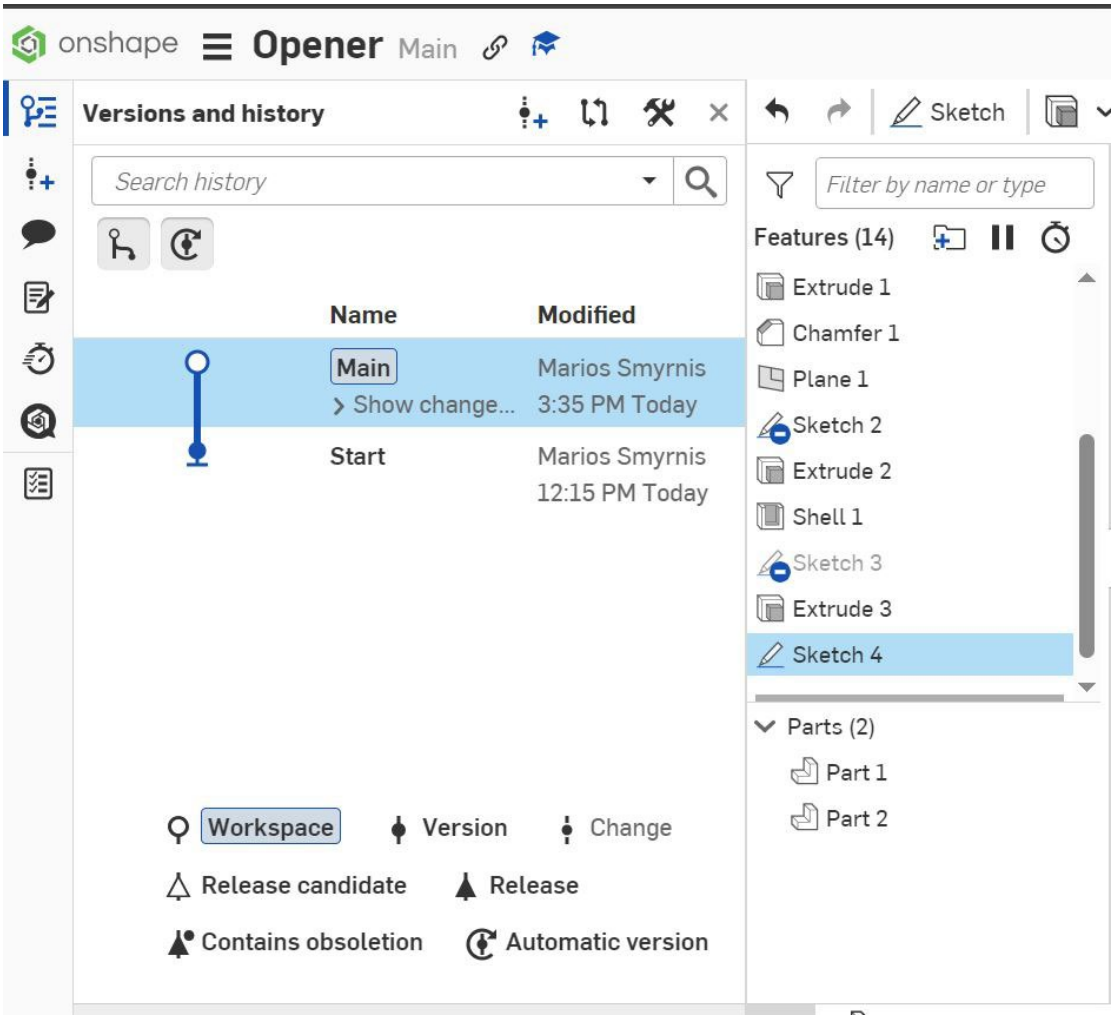
2.14.3.14. Οδηγός Τεχνητής Νοημοσύνης Onshape

Επιπλέον έχει δυνατότητα για να κάνει έλεγχο του αριθμού εντολών που δίνονται από τον χρήστη, καταμέτρηση του χρόνου εργασίας, αριθμού σχεδίων, συναρμολογήσεων (assemblies), γραφικών απεικόνισης σχεδίου, παρόμοιο με τεστ Benchmark:

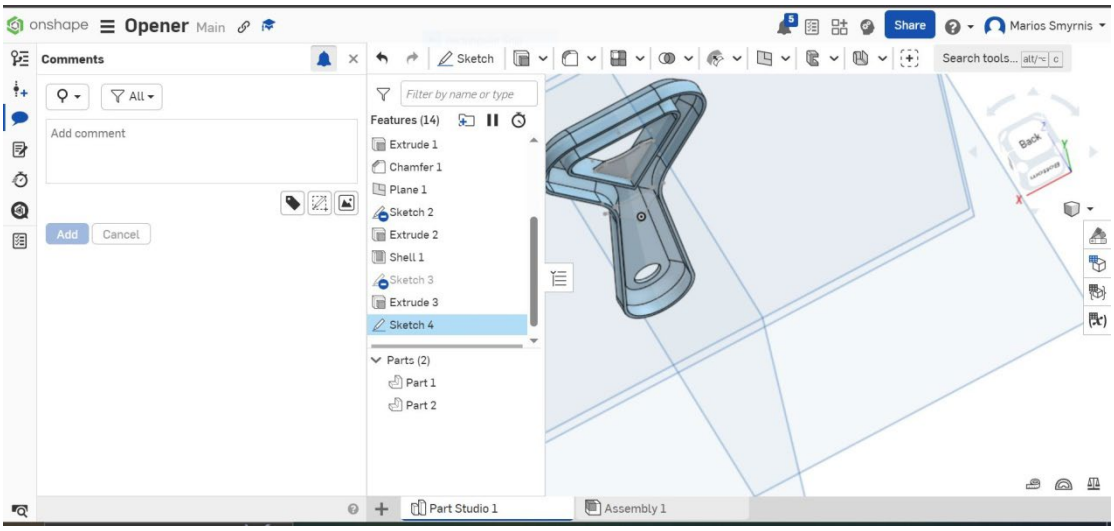


2.14.3.15. Αναφορά Απόδοσης Onshape

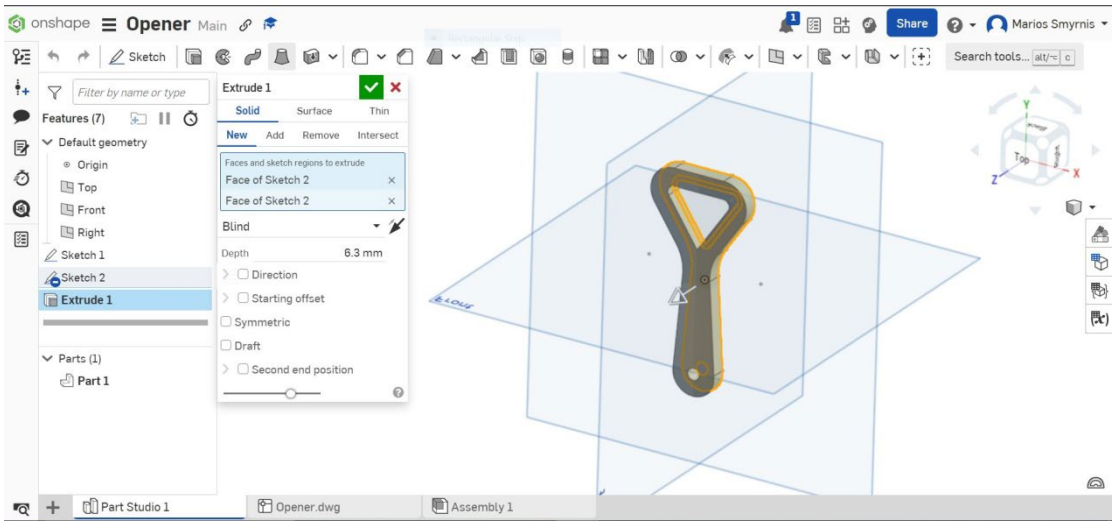
Το πρόγραμμα διαθέτει ιστορικό για πότε ξεκίνησε κάποιο έργο από τον χρήστη και αναλόγως την επιθυμία του χρήστη να δρομολογηθεί μέχρι την τελική του μορφή. Επειδή όλα τα σχέδια διατίθενται στο διαδίκτυο και είναι ελεύθερα προς όλους τους χρήστες της πλατφόρμας, επίσης, δίνεται η δυνατότητα της εισαγωγής σχολίων ως προς τα σχέδια του ιδίου, ή άλλων χρηστών:



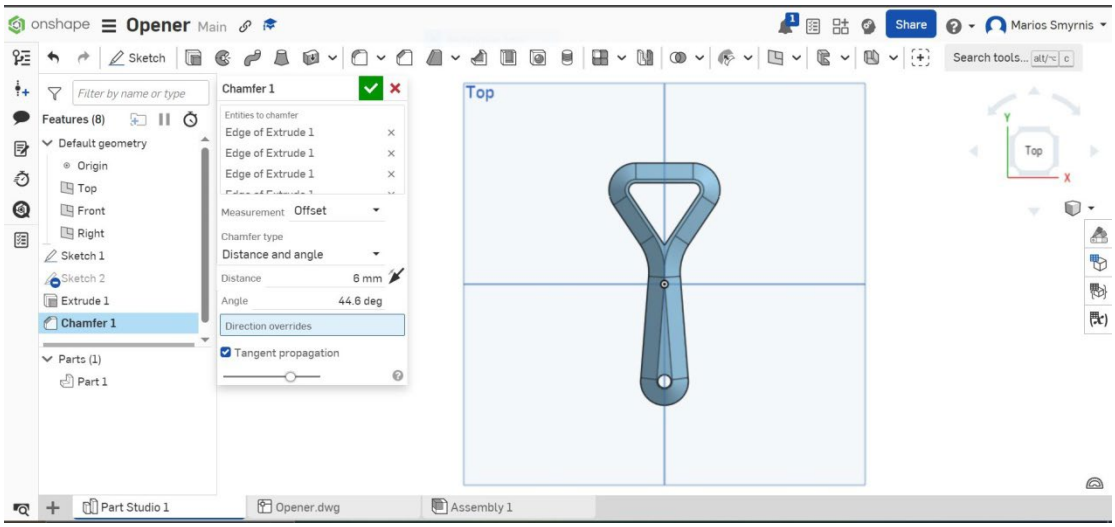
2.14.3.16. Ιστορικό Onshape



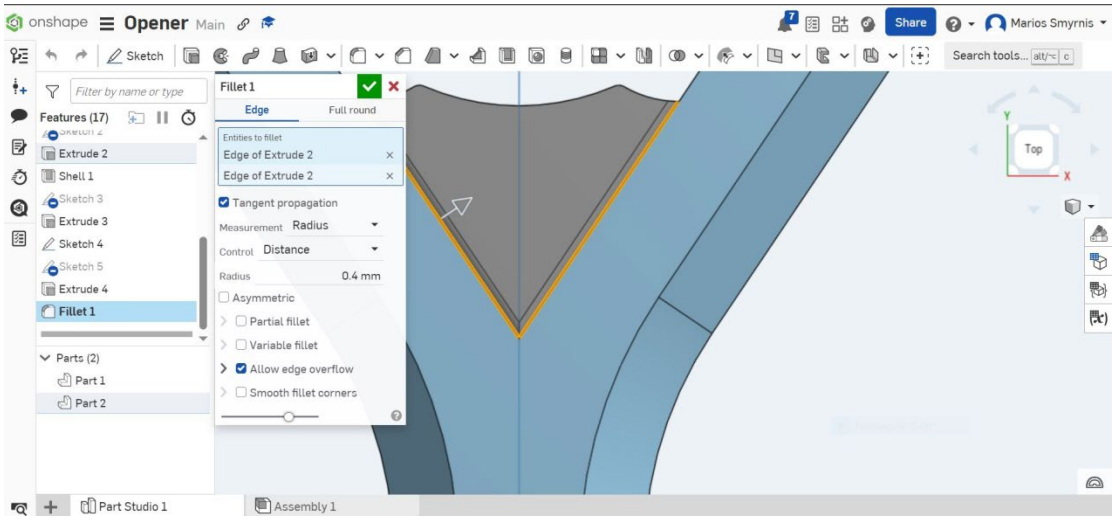
2.14.3.17. Πλαίσιο Σχολιασμού Onshape



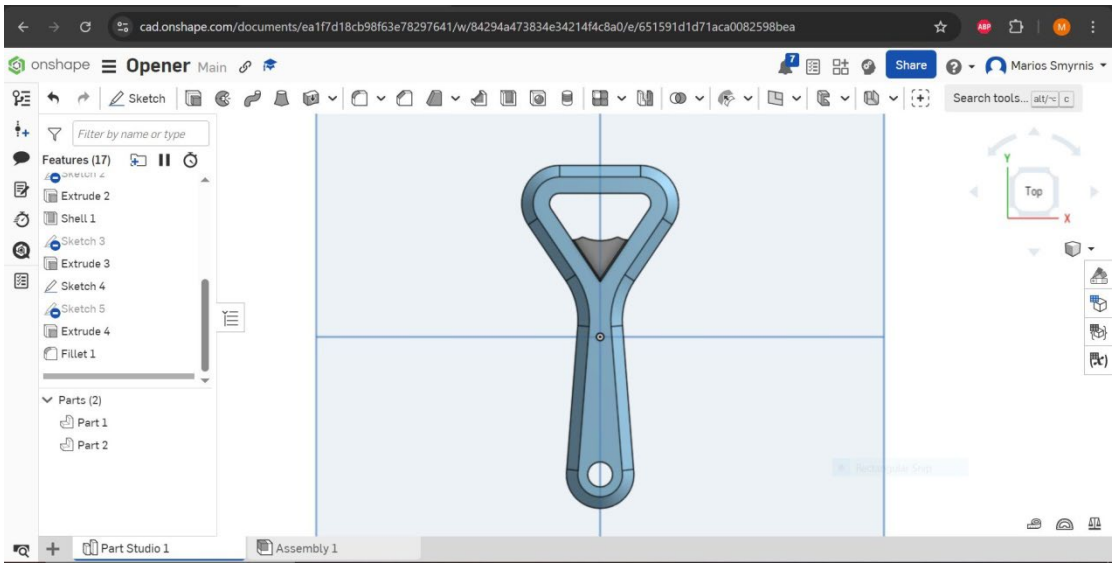
2.14.3.18. Εντολή εξώθησης Onshape



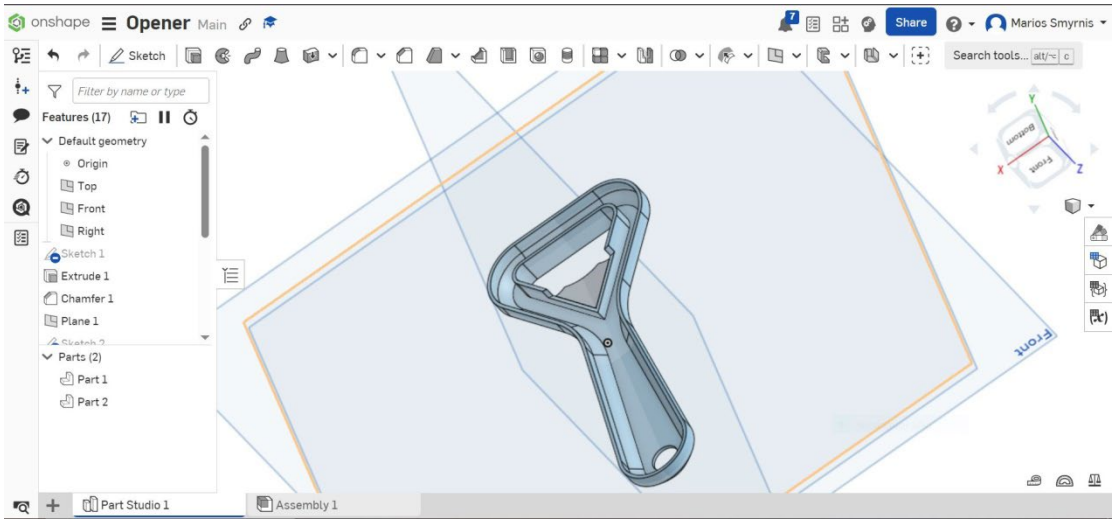
2.14.3.19. Εντολή Chamfer Onshape



2.14.3.20. Εντολή στρογγυλέματος (round) Onshape



2.14.3.21. Πρόοψη Onshape



2.14.3.22. Πίσω όψη Onshape

Autodesk Fusion student edition: Το Autodesk Fusion (πρώην Fusion 360) είναι μια εμπορική εφαρμογή λογισμικού σχεδιασμού με τη βοήθεια υπολογιστή (CAD), κατασκευής με τη βοήθεια υπολογιστή (CAM), μηχανικής με τη βοήθεια υπολογιστή (CAE) και σχεδίασης πλακέτας τυπωμένων κυκλωμάτων (PCB), που αναπτύχθηκε από την Autodesk. Είναι διαθέσιμο για Windows, macOS και προγράμματα περιήγησης ιστού, με απλοποιημένες εφαρμογές μόνο για προβολή διαθέσιμες για Android και iOS. Το Fusion διατίθεται με άδεια χρήσης ως συνδρομή επί πληρωμή, με διαθέσιμη μια δωρεάν περιορισμένη έκδοση για οικιακή χρήση, μη εμπορική, προσωπική χρήση.

Το Fusion παρουσιάστηκε από την Autodesk στις 24 Σεπτεμβρίου 2013. Ενσωματώνει πολλά χαρακτηριστικά του Inventor Fusion, το οποίο αντικαθιστά.

- Το 2009, κυκλοφόρησε η τεχνολογική επίδειξη Inventor Fusion.

- Το καλοκαίρι του 2013, το Fusion 360 ανακοινώθηκε δημόσια ως μια έκδοση του αρχικού με δυνατότητα cloud.

- Τον Ιανουάριο του 2024, το Fusion μετονομάστηκε, αφαιρώντας το '360' από το προηγούμενο όνομα προϊόντος 'Fusion 360'.

Μετά την κυκλοφορία, άλλα προϊόντα της Autodesk ενσωματώθηκαν στο Autodesk Fusion:

- Το 2017, ενσωματώθηκε η λειτουργία Slicer του Autodesk 123D.

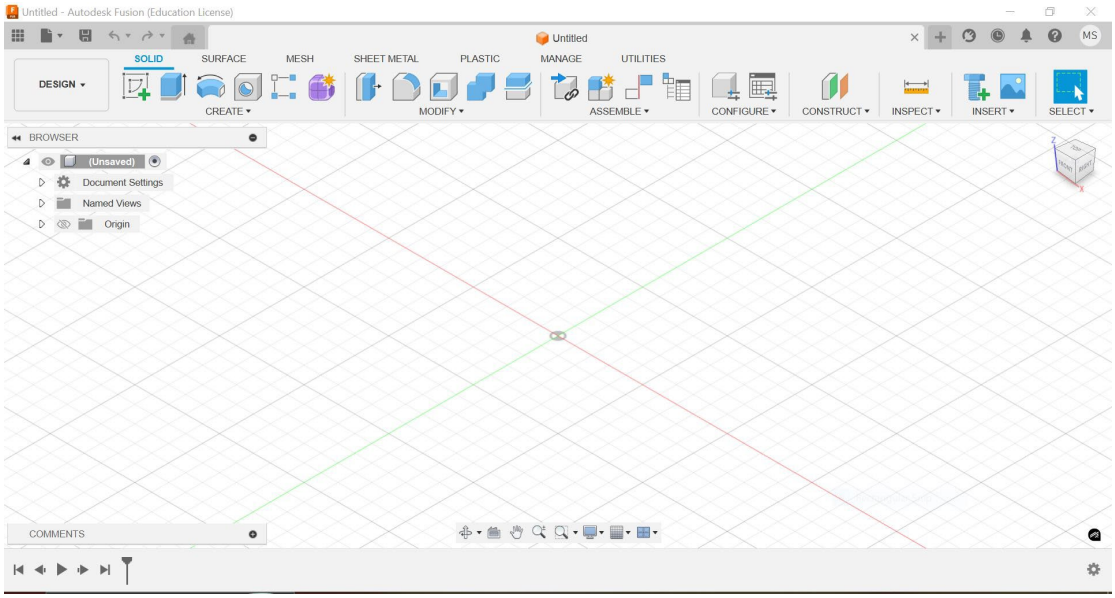
- Το 2021, το Autodesk Meshmixer διακόπηκε, αφού ενσωματώθηκε η λειτουργικότητα στο Fusion 360.

- Το 2021, το Netfabb συγχωνεύτηκε με το Fusion 360.

Το Fusion διαθέτει ενσωματωμένες δυνατότητες για τρισδιάστατη μοντελοποίηση, συνεργασία, προσομοίωση και τεκμηρίωση. Μπορεί να διαχειρίζεται διαδικασίες κατασκευής όπως κατεργασία, φρεζάρισμα, τόρνευση και προσθετική κατασκευή. Διαθέτει επίσης λειτουργίες αυτοματισμού ηλεκτρονικού σχεδιασμού (EDA), όπως σχηματικό σχεδιασμό, σχεδιασμό PCB και διαχείριση εξαρτημάτων. Μπορεί επίσης

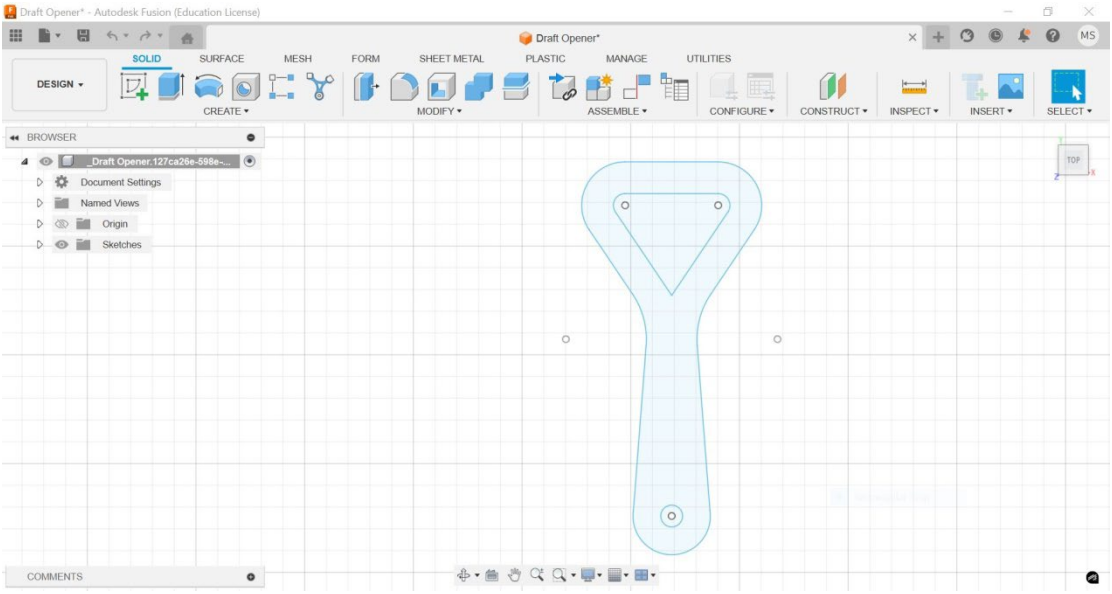
να χρησιμοποιηθεί για απεικόνιση, animation, γενετικό σχεδιασμό και μια σειρά από προηγμένες εργασίες προσομοίωσης (FEA).

Το Fusion χρησιμοποιεί μια σειρά από λειτουργίες τεχνητής νοημοσύνης που βασίζονται στο Cloud, π.χ. για προβολές σχεδίασης, αυτοματοποιημένη μοντελοποίηση, γενετικό σχεδιασμό, περιορισμούς σκίτσων.



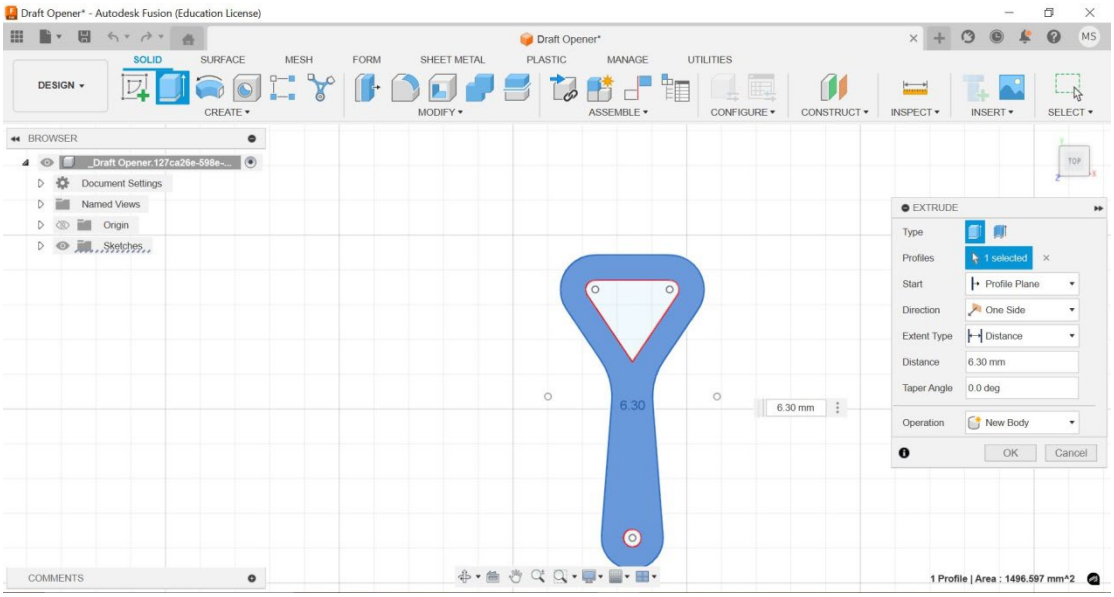
2.14.3.23. Διεπαφή του Fusion

Σε αντίθεση με το Onshape, το Fusion προαπαιτεί την αποθήκευση και χρήση διαβιβαστή (client) για πρόσβαση στο cloud, αντί συνδέσμου με απευθείας σύνδεση μέσω email το οποίο παρέχει το πρώτο. Επιπλέον ενώ οι λειτουργίες στο cloud είναι αρκετά κοινές, το Fusion χρειάζεται αποθήκευση της εργασίας του χρήστη, ειδάλλως, σε περίπτωση αποσύνδεσης με την πλατφόρμα η εργασία αυτή χάνεται.



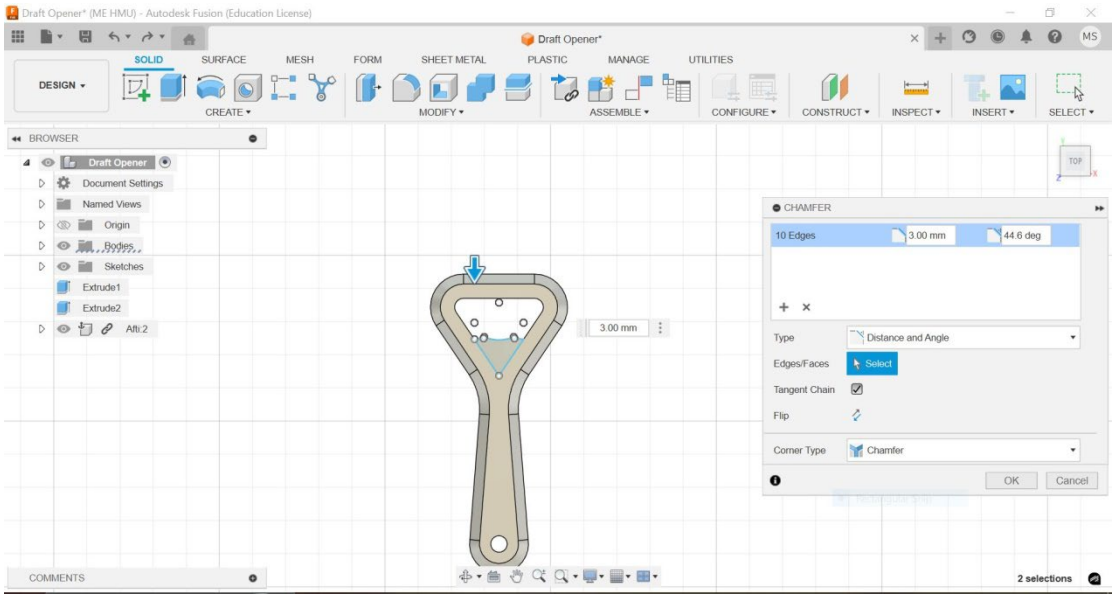
2.14.3.24. Δισδιάστατη σχεδίαση στο Fusion

Το Fusion χρησιμοποιεί κόμβο (hub) στο Cloud διαμέσου του οποίου ο χρήστης μπορεί να ανεβάσει, να δημιουργήσει και να δημοσιεύσει τα σχέδια του. Μπορεί επίσης να καλέσει άλλους χρήστες του Fusion, που έχουν τα δικά τους hub να σχολιάσουν και να επεξεργαστούν τα σχέδια, μέσω πρόσκλησης. Στην περιορισμένη student edition, όμως, κάτι τέτοιο δεν είναι εφικτό για λόγους προστασίας πνευματικών δικαιωμάτων.

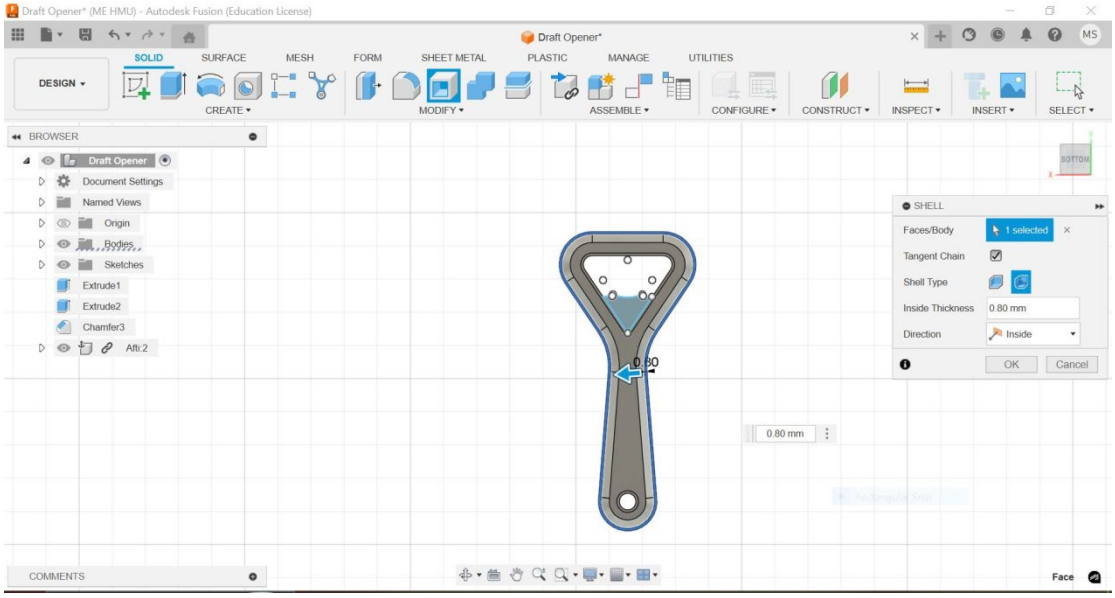


2.14.3.25. Εξώθηση (extrude) στο Fusion

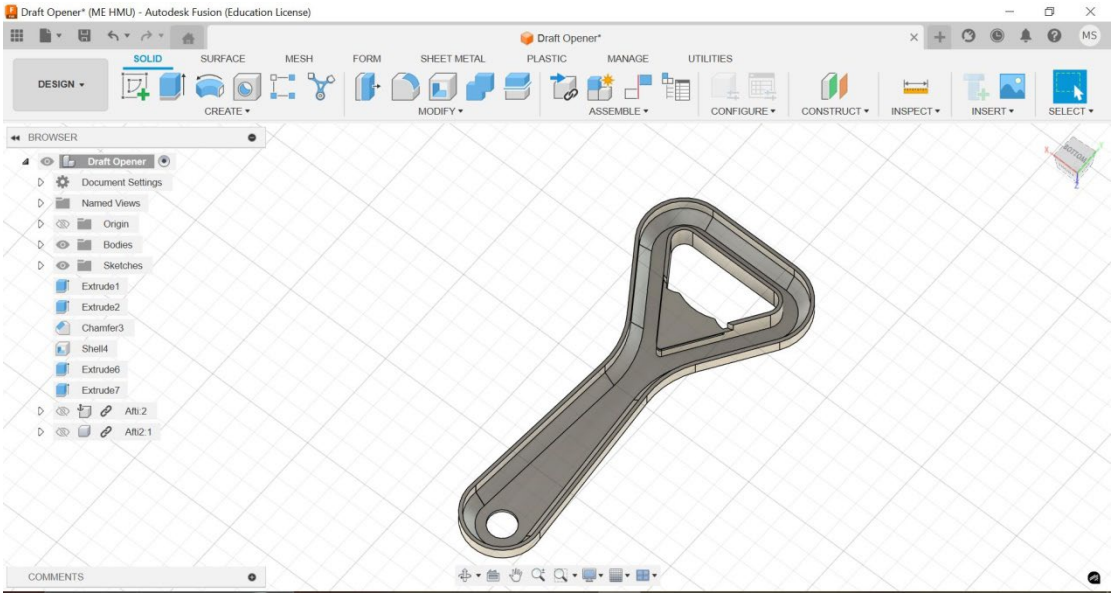
Το Fusion χρησιμοποιεί ιστορικό σχεδιασμού, όπως τα περισσότερα 3D CAD λογισμικά με τις λειτουργίες και τις εντολές σχεδιασμού και μορφοποίησης να είναι επίσης παρόμοιες με την πλειοψηφία των αντίστοιχων προγραμμάτων σχεδίασης. Ωστόσο, δεν έχει τη δυνατότητα τροποποίησης (edit) προηγούμενων εντολών στο ιστορικό του, με αποτέλεσμα να έχει μια ψευδο αναδημιουργόμενη μορφή σχεδιασμού, η οποία όμως πάει κόντρα στο σκοπό του κατασκευαστή (designer's intent).



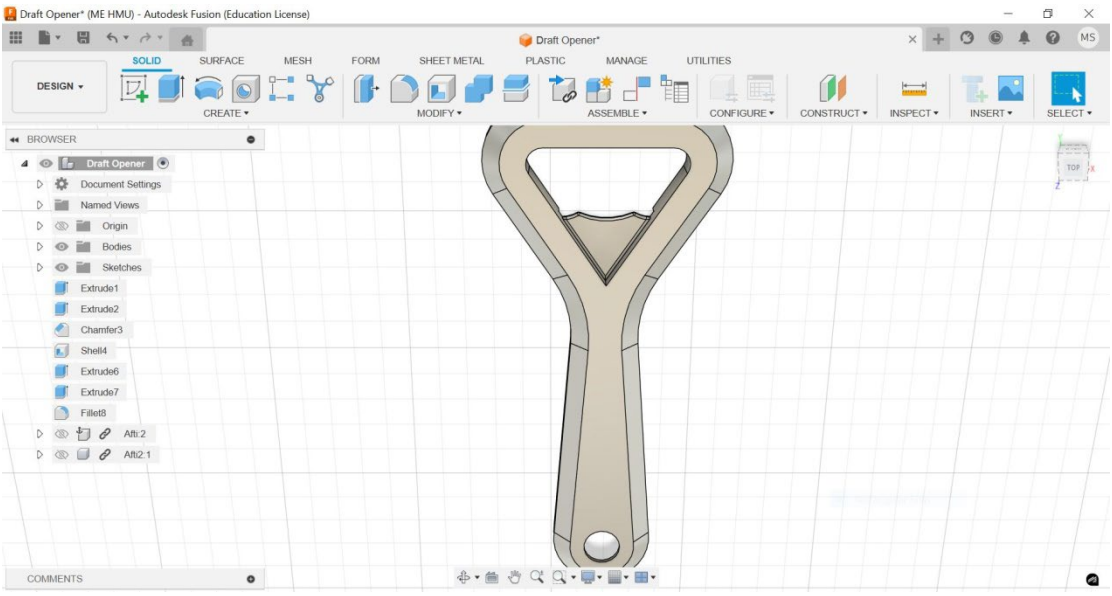
2.14.3.26. Εντολή Chamfer στο Fusion



2.14.3.27. Εντολή κουφώματος (shell) στο Fusion



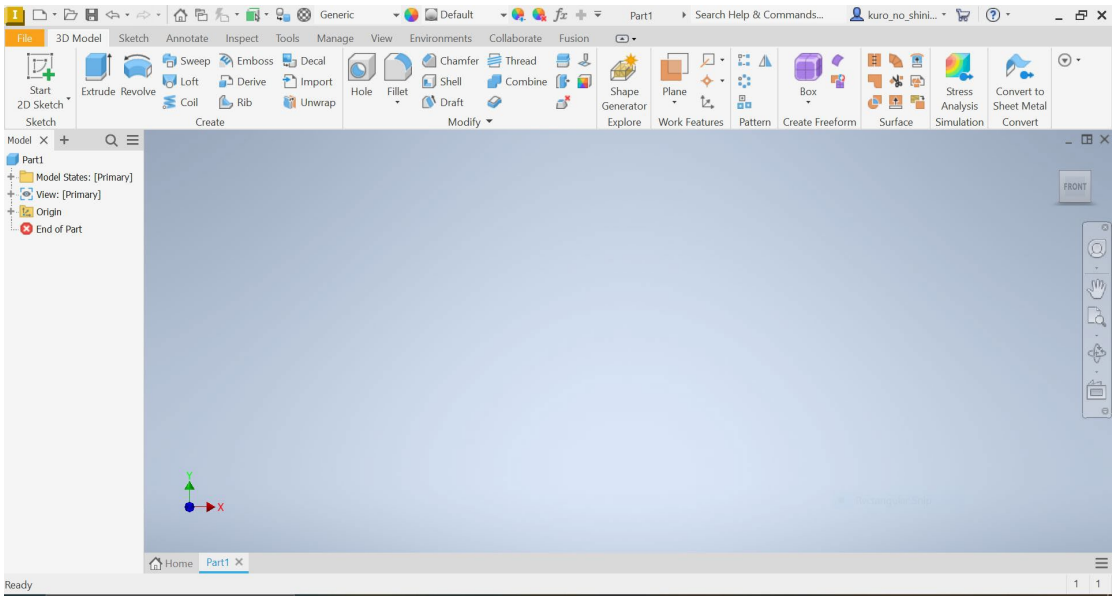
2.14.3.28. Πίσω όψη στο Fusion



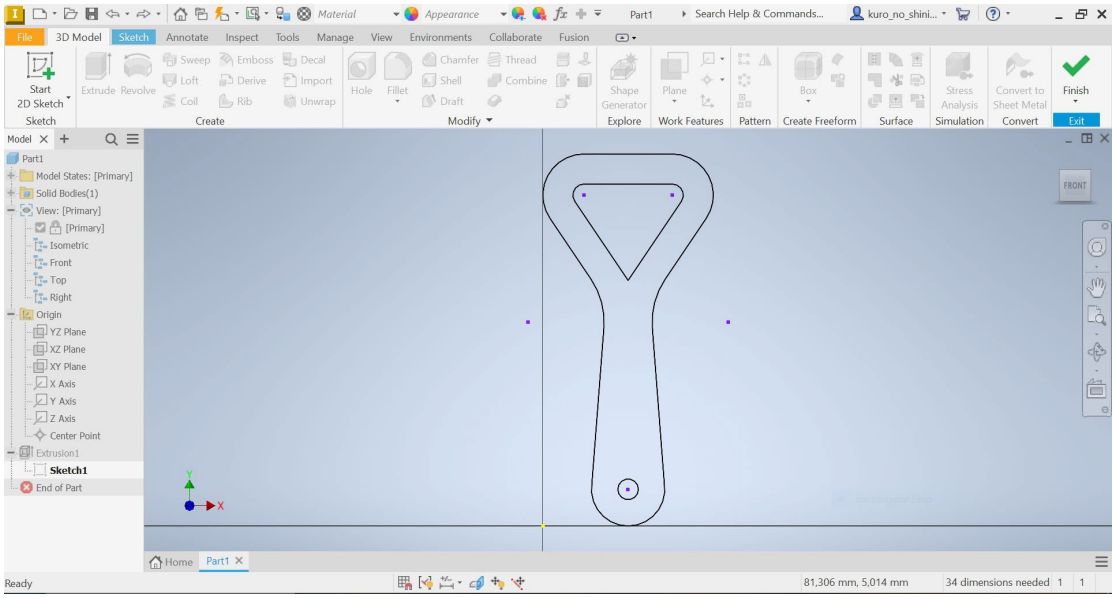
2.14.3.29. Πρόοψη στο Fusion

Autodesk Inventor student edition: Το Autodesk Inventor είναι μια εφαρμογή επέκτασης σχεδιασμού με τη βοήθεια υπολογιστή για τρισδιάστατο μηχανολογικό σχεδιασμό, προσομοίωση, οπτικοποίηση και τεκμηρίωση που αναπτύχθηκε από την Autodesk.

Το Inventor επιτρέπει την ανάλυση δισδιάστατων και τρισδιάστατων μοντέλων CAD δημιουργώντας μια εικονική αναπαράσταση του τελικού προϊόντος, προκειμένου να επικυρωθεί η μορφή, η εφαρμογή και η λειτουργία πριν από την κατασκευή του. Το Autodesk Inventor υποστηρίζει παραμετρική, άμεση επεξεργασία, μοντελοποίηση εξαρτημάτων ελεύθερης μορφής και μοντελοποίηση συναρμολόγησης. Παρέχει υποστήριξη για διάφορες μορφές CAD και τυπικά σχέδια DWG.

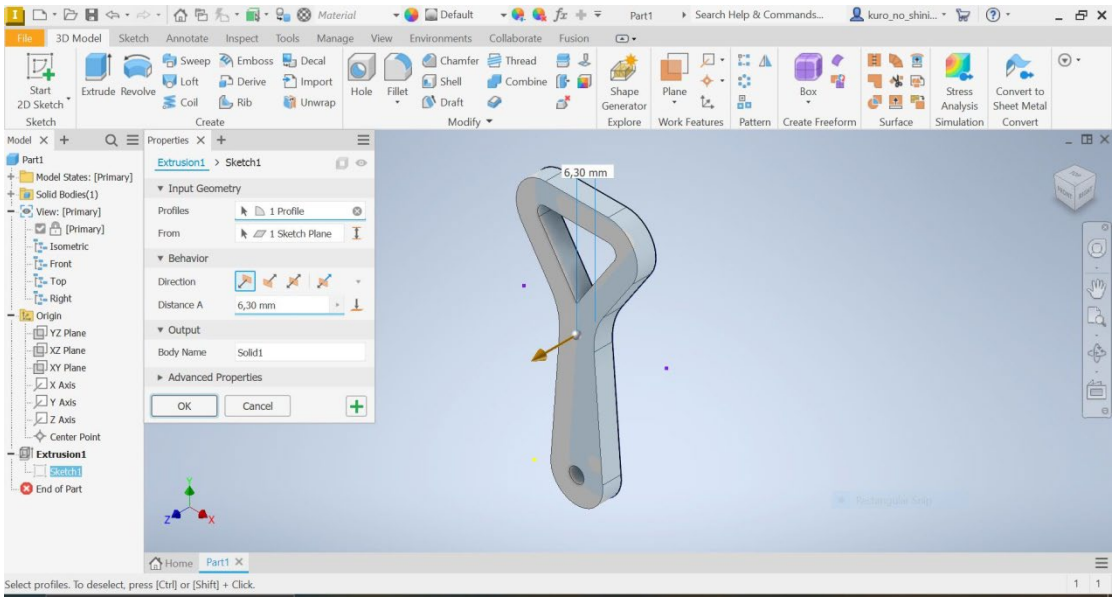


2.14.3.30. Διεπαφή του Inventor

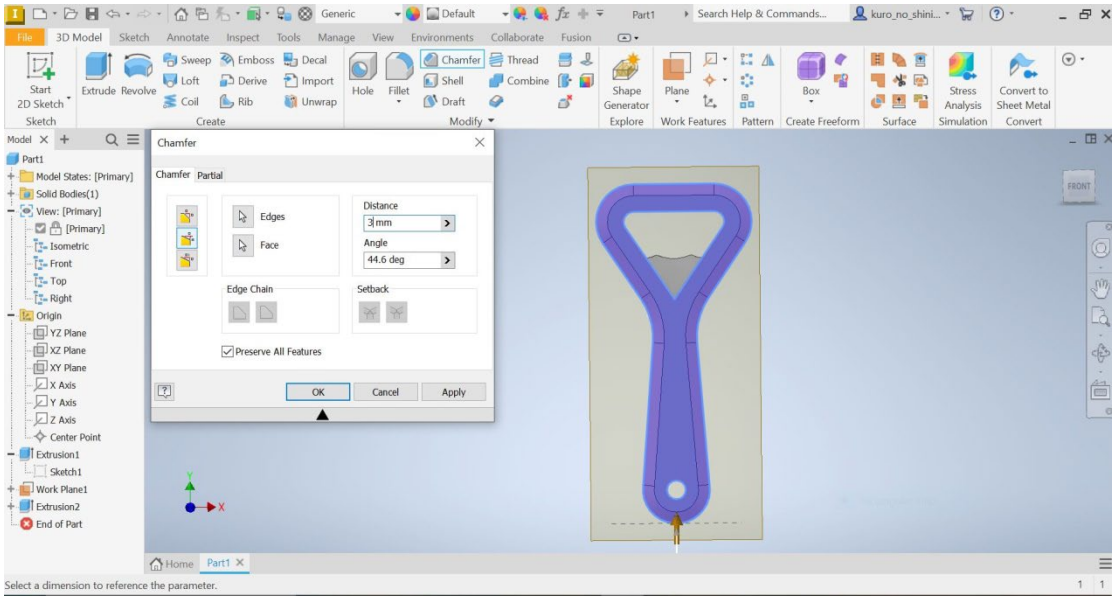


2.14.3.31. Δυσδιάστατη απεικόνιση στο Inventor

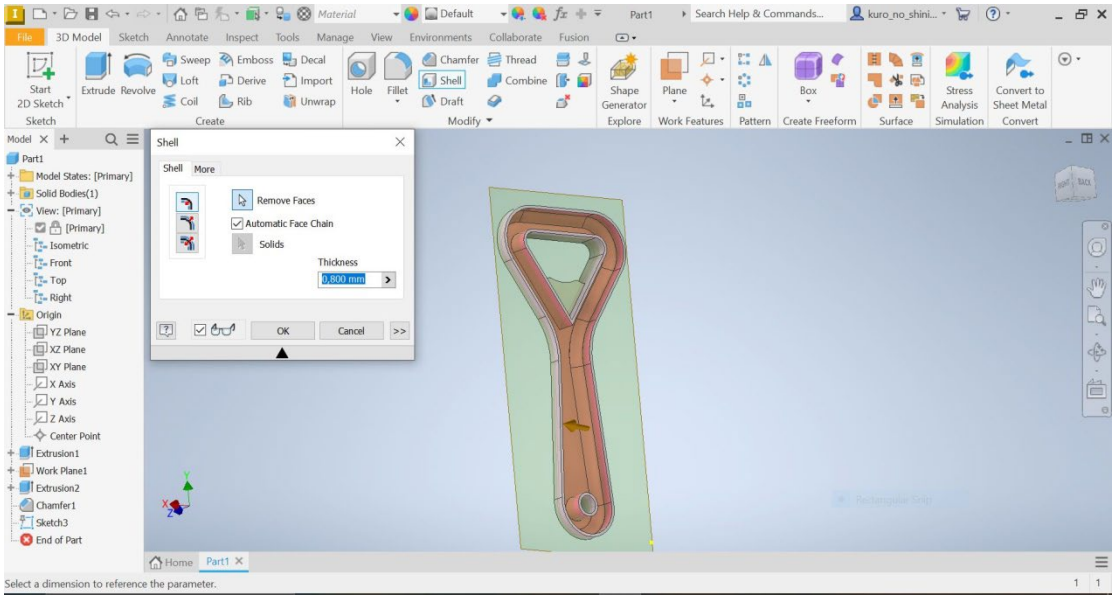
Το Inventor λειτουργεί με παρόμοιο τρόπο όπως το Creo Parametric στο σχεδιασμό σκίτσου, συγκρατήσεων και γεωμετρικών εντολών. Η τρισδιάστατη σχεδίαση του επίσης έχει μεγάλες ομοιότητες σαν δυνατότητες σχεδιασμού κι επιλογών. Οι επιπλέον δυνατότητες που έχει είναι η βιβλιοθήκη που εμπεριέχεται στο λογισμικό σαν έξτρα επιλογή, καθώς επίσης και λίστα με έτοιμα πρότυπα σχέδια, DIN, ANSI κ.α. και μια ηλεκτρονική λίστα από ηλεκτρολογικές εγκαταστάσεις, η οποία αφορά τον ηλεκτρολογικό σχεδιασμό με το λογισμικό.



2.14.3.32. Εντολή εξώθησης (extrude) στο Inventor

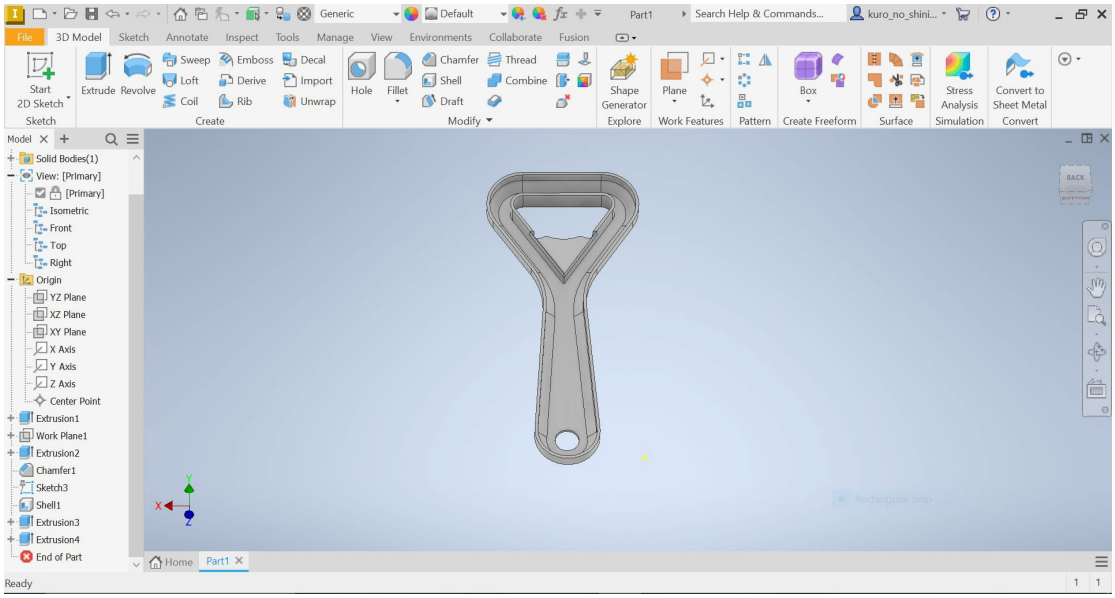


2.14.3.33. Chamfer στο Inventor

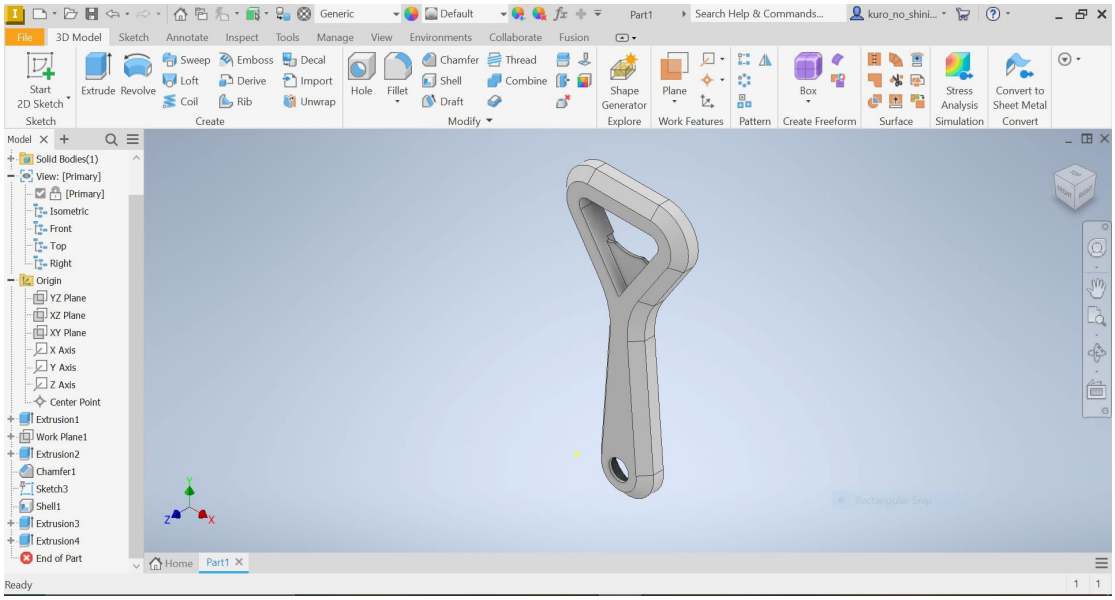


2.14.3.34. Εντολή κουφώματος (shell) στο Inventor

Μια έξτρα επιλογή που έχει το Inventor ως πρόγραμμα που έχει αναπτυχθεί από την Autodesk, είναι η άμεση συσχέτιση και τροποποίηση ενός σχεδίου από το Autocad στο Inventor. Όταν εισαχθεί ένα σχέδιο από το Autocad, μέσω της επιλογής εισαγωγής (Import) στο Inventor και μόλις εφαρμοστεί η εντολή από την καρτέλα Εργαλείων (Tools), τότε μπορεί να δημιουργηθεί μια συσχετιστική σχέση μεταξύ των 2 προγραμμάτων. Ειδικότερα, μετά την εισαγωγή του σχεδίου στο Inventor, επιλέγοντας την εντολή Project Geometry πάνω στο σκίτσο, τότε ότι επεξεργασία υποστεί το ίδιο σχέδιο στο Autocad θα συμβεί και στην δισδιάστατη απεικόνιση του στο Inventor.



2.14.3.35. Πίσω όψη στο Inventor



2.14.3.36. Μπροστινή όψη στο Inventor

2.14.4. Σύγκριση και αξιολόγηση λογισμικών

Στην προηγούμενη ενότητα παρουσιάστηκαν τα λογισμικά ελευθέρας χρήσης, τα οποία χρησιμοποιήθηκαν για τον σχεδιασμό ενός κοινού σχεδίου (Εικ. 2.14.3.1.), σε σχέση με το Autocad και το Creo Parametric και σειρά έχει η σύγκριση και η αξιολόγηση του καθενός πάνω σε αυτό τον τομέα:

- **Creo Elements/Direct Modeling:** Το Creo/Elements Direct είναι στην πραγματικότητα το CoCreate Modeling, ένα ολοκληρωμένο σύστημα μοντελοποίησης 3D CAD Direct (χωρίς ιστορικό) υψηλής τεχνολογίας. Περιλαμβάνει τη μονάδα σχολιασμού για τη δημιουργία δισδιάστατων σχεδίων που σχετίζονται με τα τρισδιάστατα μοντέλα και τα συστήματα. Ενώ το Creo Parametric είναι ιδανικό για παραμετρική μοντελοποίηση, το Creo/Elements Direct είναι ιδανικό για ανάπτυξη, ειδικά όταν δεν είναι σίγουρο πού θα καταλήξει το εν ενεργεία σχέδιο. Το Creo/Direct είναι στην πραγματικότητα μια ελαφρύτερη μηχανή του Creo Parametric με ένα περιβάλλον χρήστη που κρύβει το ιστορικό σχεδίασης. Έτσι, οι εντολές λειτουργούν όπως οι εντολές άμεσης μοντελοποίησης, αλλά στην πραγματικότητα, δημιουργείται ένα ιστορικό. Όταν ανοίγει ένα μοντέλο Creo/Direct στο Creo/Parametric, φαίνεται αυτό το ιστορικό. Αυτό σημαίνει ότι αν κάποιος χρησιμοποιεί το Creo/Direct με ακατάστατο τρόπο, θα καταλήξει με ένα άχρηστο κομμάτι. Στο Creo/Elements Direct δεν υπάρχει τέτοιο ιστορικό, επομένως ο χρήστης δεν χρειάζεται να σκεφτεί στρατηγικές μοντελοποίησης. Έτσι, το Creo/Direct απευθύνεται σε χρήστες που δεν χρησιμοποιούν CAD, οι οποίοι μπορούν να κάνουν απλές τροποποιήσεις σε μοντέλα Creo Parametric με αυτό. Για σοβαρές τροποποιήσεις σε μοντέλα Pro Engineer, η Επέκταση Ευέλικτης Μοντελοποίησης του Creo Parametric είναι η καλύτερη επιλογή, επειδή επιτρέπει ισχυρή άμεση μοντελοποίηση χωρίς να προκαλεί προβλήματα στο ιστορικό σχεδίασης.
- **Solid Edge student edition:** Θεωρητικά το μεγαλύτερο πλεονέκτημα του Solid Edge είναι η σύγχρονη μοντελοποίηση. Βασικά, σε αντίθεση με την παραδοσιακή παραμετρική μοντελοποίηση όπου υπάρχει ένα ιστορικό σχεδίασης με ένα άκαμπτο δέντρο και η επεξεργασία σε κάτι από επάνω μπορεί να έχει τεράστιες επιπτώσεις σε πράγματα πιο κάτω, το Synchronous επιτρέπει να αντιμετωπίζεται ολόκληρο το μοντέλο ως μία ενιαία οντότητα και να επεξεργάζεται οποιαδήποτε διάσταση σε αυτό δίχως αλλαγές στις άλλες τρέχουσες διαστάσεις. Απαλλάσσει από το ιστορικό. Στην πράξη, ωστόσο, είναι πάρα πολύ περιορισμένη. Πολλά εργαλεία δεν λειτουργούν σε σύγχρονη λειτουργία. Το να υπάρχει περισσότερη ισχύ επεξεργασίας εν κινήσει δεν έχει σημασία αν δεν γίνεται αρχικά να δημιουργηθεί το σχήμα που χρειάζεται. Από την άλλη, το Solid Edge διαθέτει ένα εξαιρετικό περιβάλλον σχεδίασης. Δεν είναι μόνο εύκολο να δημιουργηθεί οποιαδήποτε λεπτομέρεια, διακεκομμένη όψη, τομή, ή διάσταση, αλλά μπορεί επίσης να γίνει βελτίωση σε όλες τις ρυθμίσεις και τα πρότυπά, ώστε να φαίνονται πραγματικά καθαρά και ευπαρουσίαστα. Η διεπαφή είναι επίσης αρκετά προσαρμόσιμη σε γενικές

γραμμές. Μόλις ρυθμιστούν τα μενού και τα πλήκτρα πρόσβασης σε κάθε περιβάλλον, θα χρειάζονται πολύ λιγότερα κλικ. Υπάρχουν επίσης πολλές καλές μακροεντολές που μπορούν να βελτιστοποιήσουν τη ροή εργασίας.

- **PTC Onshape**: Είναι ένα λογισμικό cloud, δηλαδή επιτρέπει την κοινή επεξεργασία σε πραγματικό χρόνο με τα μέλη μιας ομάδας, εξαλείφοντας τα προβλήματα ελέγχου εκδόσεων. Προσβασιμότητα σε πρόγραμμα περιήγησης ιστού, που σημαίνει ότι μπορεί να χρησιμοποιηθεί από σχεδόν οποιαδήποτε συσκευή και δεν απαιτείται εγκατάσταση. Όλη η εργασία αποθηκεύεται αυτόματα στο cloud, επομένως δεν υπάρχουν ανησυχίες για απώλεια δεδομένων από βλάβη υλικού, ή σφάλματα. Αυτό βέβαια έχει τα ρίσκα του δεδομένου πόσο επιρρεπές είναι το διαδύκτυο σε ιούς, ή επιθέσεις από hackers και με την πρόσφατη ιστορία μεγαλοεταιριών να διαρρέρουν υλικό που κανονικά υπόκειται σε προστασία δεδομένων. Έχει ενσωματωμένες λειτουργίες Διαχείρισης Δεδομένων Προϊόντος (PDM), όπως ο έλεγχος εκδόσεων και ένα πλήρες ιστορικό επεξεργασιών, περιλαμβάνονται χωρίς επιπλέον κόστος. Επιπλέον δεν έχει καμία εξάρτηση από hardware: Οι χρήστες μπορούν να το χρησιμοποιούν σε λιγότερο ισχυρούς υπολογιστές, καθώς η επεξεργασία γίνεται στο cloud. Τα αρνητικά του είναι ότι απαιτεί μόνιμη πρόσβαση στο διαδίκτυο ώστε να λειτουργήσει. Επίσης διαθέτει περιορισμένα χαρακτηριστικά και λειτουργικές εφαρμογές σε σχέση με το εμπορικό προϊόν της ίδιας εταιρείας.
- **Autodesk Fusion student edition:** Όπως και το Onshape είναι 1 λογισμικό μορφής cloud που σημαίνει ότι πολλά από αυτά που ισχύουν για το ένα θα ισχύουν και για το άλλο. Μια διαφορά του Fusion είναι ότι δεν λειτουργεί αυτόματα σε οποιοδήποτε περιηγητή ιστού όπως το Onshape, αλλά χρειάζεται διαβιβαστή (client) ώστε να έχει πρόσβαση στο cloud και η πορεία εργασίας δεν αναβαθμίζεται/αποθηκεύεται αυτόματα, αλλά πρέπει να αποθηκεύεται συχνά από τον διαβιβαστή ώστε να μην υπάρξει ρίσκο να χαθεί δουλειά. Επιπροσθέτως, μέσω του κόμβου (hub) που ανοίγει ο χρήστης όταν ξεκινάει κάποια εργασία έχει την δυνατότητα να προσκαλέσει άλλους χρήστες σε διαφορετικά hub για να σχολιάσουν/αξιολογήσουν την δουλειά που γίνεται. Ενώ διαθέτει περισσότερες επιλογές και χαρακτηριστικά από το Onshape και τα 2 δεν έχουν πολύ φιλική προς τον χρήστη σχέση με τις εντολές τους, ιδίως αν ο χρήστης δεν είναι πολύ εξοικειωμένος στην χρήση των CAD. Τέλος, δεν υπάρχει δυνατότητα τροποποίησης περασμένων εντολών, καθιστώντας το δύσχρηστο για επεξεργασία πολύπλοκης σχεδίασης.
- **Autodesk Inventor student edition:** Το Inventor έχει μια πολύ φιλική προς τον χρήστη περιβάλλον εργασίας με ένα πλήθος λειτουργιών και χαρακτηριστικών τα οποία είναι πολύ εύχρηστα ως προς τον τρόπο λειτουργίας τους και εύκολα στην κατανόηση. Αρκετά παρόμοιο με το Creo Parametric, ιδίως ως λογισμικά παραμετρική μοντελοποίησης. Στον σχεδιασμό είναι επίσης εύχρηστο, καθώς είναι δυνατόν να γίνουν αλλαγές σε τυχόν διατάξεις, παραδόξως δεδομένου ότι έχει αρκετούς περιορισμούς σαν πρόγραμμα. Συγκεκριμένα, υστερεί στο κομμάτι των συγκρατήσεων σκίτσου (restraints) και μόνο σε αυτόν τον τομέα γίνεται πολύ περιοριστικό.

2.14.5 Εφαρμογές

Στον μηχανολογικό σχεδιασμό τα έγγραφα κειμένου είναι κανόνας στις περισσότερες βιομηχανίες, και η μορφή καθορίζεται εμφανής, συνήθως χρησιμοποιώντας συμβατικό λογισμικό CAD. Για τον σχεδιασμό προϊόντων στον βιομηχανικό σχεδιασμό και στην αρχιτεκτονική, χρησιμοποιούνται συχνά σκίτσα και σχόλια που χρησιμοποιούνται συνήθως για να καταγράψουν το πνεύμα, ή την ουσία ενός σχεδίου. Αυτά μπορεί να είναι, ή να μην είναι μηχανογραφημένα. Εντούτοις, σε όλα αυτά τα πεδία, τα διαγράμματα φαίνεται σπάνια να χρησιμοποιούνται ως "επίσημες" αναπαραστάσεις πρώιμων πληροφοριών σχεδιασμού. Αυτό είναι οξύμορο, δεδομένης της τάσης των σχεδιαστών να είναι «οπτικοί στοχαστές». Ωστόσο όσον αφορά τη συνολική αξία, τα διαγράμματα είναι πλούσια αντικείμενα γύρω από τα οποία μπορεί κανείς να φέρει σε επαφή ενδιαφερόμενους φορείς, πράγμα που είναι καθοριστικό για έναν αποτελεσματικό και αειφόρο σχεδιασμό (π.χ. το Σπίτι της Ποιότητας).

Στερεά Μοντελοποίηση

Αυτή η διαδικασία χρησιμοποιείται για τη δημιουργία στερεών συστατικών του επιθυμητού σχήματος συνδέοντας και κόβοντας διαφορετικούς στερεούς όγκους. Το τελικό στερεό μοντέλο είναι ένα εικονικό αντίγραφο ενός πραγματικού προϊόντος, αλλά μπορεί να παρουσιαστεί και να περιστραφεί σαν ένα πραγματικό προϊόν. Υπάρχουν δύο κύριοι τύποι:

- Άμμεση, όπου το μοντέλο μπορεί να επεξεργαστεί με αναίρεση ή τροποποίηση του μοντέλου, απευθείας σε τρεις διαστάσεις.

- Παραμετρική, όπου το μοντέλο κατασκευάζεται χρησιμοποιώντας παραμέτρους (μεταβλητές ποσότητες, όπως μετρήσεις) και το μοντέλο μπορεί να επεξεργαστεί σε οποιοδήποτε σημείο του ιστορικού του.

Μοντελοποίηση επιφανείας

Αυτή η διαδικασία χρησιμοποιείται για τη δημιουργία επιφανειών επιθυμητού σχήματος με το κόψιμο, τη ραφή και την ένωση διαφορετικών επιφανειών για να δημιουργηθεί ένα τελικό μοντέλο σχήματος.

Συνέλευση

Αυτή η διαδικασία χρησιμοποιείται για τη συναρμολόγηση των μοντέλων που δημιουργήθηκαν σε στερεά ή μοντέλα επιφάνειας για να δημιουργηθεί μια τελική συναρμολόγηση. Αυτό χρησιμοποιείται ώστε να φανεί η πραγματική τοποθέτηση όλων των μοντέλων, καθώς και η πραγματική λειτουργία της συναρμολόγησης.

Σχεδίαση λεπτομερειών

Αυτή η διαδικασία χρησιμοποιείται για τη δημιουργία των 2D σχεδίων των εξαρτημάτων ή συναρμολογιών. συνήθως απευθείας από ένα 3D μοντέλο, αν και το 2D CAD μπορεί να δημιουργήσει απευθείας σχέδια 2D.

Αντίστροφη μηχανική

Αυτή η διαδικασία χρησιμοποιείται για τη μετατροπή του πραγματικού στοιχείου σε μοντέλο 3D CAD. Διαφορετικοί τύποι οργάνων όπως ο σαρωτής λέιζερ, ο σαρωτής λευκού φωτός, ο μηχανισμός μέτρησης συντεταγμένων CMM κ.λ.π. χρησιμοποιούνται για τη μέτρηση, ή την αποκωδικοποίηση του σχήματος του πραγματικού μοντέλου. Το εικονικό μοντέλο που αποκτάται μπορεί στη συνέχεια να τροποποιηθεί σε ένα πρόγραμμα CAD.

Designing / Cutting

Χρησιμοποιώντας λογισμικό 2D CAD ή γραφικών, αυτά τα μηχανήματα μπορούν να σχεδιάσουν ή να κόψουν 2D σχήματα σε χαρτί, χαρτί ή αυτοκόλλητα φύλλα βινυλίου.

Κοπή λέιζερ

Χρησιμοποιώντας λογισμικό 2D CAD ή γραφικών, αυτά τα μηχανήματα μπορούν να κόψουν ή να χαράξουν μια μεγάλη ποικιλία υλικών όπως κάρτα, κόντρα πλακέ, ακρυλικό φύλλο, υφάσματα, γυαλί.

2.5D milling

Τυπικά, η επεξεργασία 2.5D χρησιμοποιεί αρχεία CAD 2D ως πηγή και περικοπές στις κατευθύνσεις x και y (παράλληλα με το κρεβάτι του μηχανήματος), ενώ χαμηλώνει προοδευτικά το εργαλείο μέσω διαφόρων βημάτων για να κόψει ένα αντικείμενο χρησιμοποιώντας ένα περιστρεφόμενο εργαλείο. Ο καθορισμός των διαδρομών εργαλείων είναι γενικά απλός και το τμήμα κοπεί γρήγορα.

3D milling

Η μηχανική κατεργασία 3D χρησιμοποιεί τυπικά αρχεία STL από προγράμματα 3D CAD ως πηγή και απαιτεί και οι τρεις άξονες του μύλου να μπορούν να κινούνται ταυτόχρονα. Η τρισδιάστατη μηχανική κατεργασία μπορεί να διαρκέσει πολύ καιρό ακόμη και σε γρήγορες και σύγχρονες μηχανές και ειδικά αν απαιτείται τελειωτικό φινίρισμα. Μπορεί επίσης να προστεθεί ένας τέταρτος, ή περιστροφικός άξονας για τη δημιουργία τμημάτων με περιστροφή.

Turning

Το λογισμικό CAM αυτοματοποιεί την παραδοσιακή περιστροφή σε ένα τόρνο, συμπεριλαμβανομένου της χονδροειδούς χύτευσης, της αύλακας, του σπειρώματος και του φινιρίσματος για ταχύτερα και ακριβέστερα αποτελέσματα.

Τρισδιάστατη εκτύπωση

Η 3D εκτύπωση είναι μια διαδικασία κατασκευής ενός τρισδιάστατου αντικειμένου σχεδόν οποιασδήποτε μορφής από ένα ψηφιακό μοντέλο. Αυτό επιτυγχάνεται με τη χρήση μιας πρόσθετης διεργασίας, όπου τοποθετούνται διαδοχικά στρώματα υλικού (συνήθως θερμοπλαστικά). Κάθε στρώμα είναι μια οριζόντια διατομή του τελικού αντικειμένου. Κάθε αντικείμενο αρχίζει με ένα αρχείο CAD, το οποίο δημιουργήθηκε με ένα πρόγραμμα 3D μοντελοποίησης, ή με αντίστροφη μηχανική.

Τα βασικά πλεονεκτήματα του CAD / CAM:

- αυξανόμενο εύρος ιδεών σχεδιασμού
- βελτιωμένη ακρίβεια
- ευκολία τροποποίησης
- επαναληψιμότητα της παραγωγής
- ποιότητα της παραγωγής
- μείωση της σπατάλης

3. Συμπεράσματα

Οι επιχειρήσεις μπορούν να εξοικονομήσουν κεφάλαιο χρησιμοποιώντας το ελεύθερο λογισμικό, όπως τα χρήματα, αναπτύσσοντας και διατηρώντας το ελεύθερο λογισμικό. Δεδομένου ότι το ελεύθερο λογισμικό δημιουργείται και συντηρείται δωρεάν, φαίνεται να συσχετίζεται η επιχειρηματική επιτυχία με την πώληση ή τη διανομή λογισμικού. Σαφώς οι εταιρείες λογισμικού επικεντρώνονται στην OSS, με την ελπίδα ότι θα αποκομίσουν έσοδα, αν και παρατάνε κάτι που προηγουμένως θα είχαν πουλήσει. Η Oracle, η Microsoft και άλλοι ιδιωτικοί πωλητές προσφέρουν εκδόσεις no-cost του λογισμικού τους, ώστε να διευκολύνουν τους προγραμματιστές και τους εφαρμοστές όλων των ειδών να επιλέξουν τις πλατφόρμες τους από αυτούς τους πωλητές. Όταν η Red Hat, η MySQL AB, η Pentaho, ή οποιοσδήποτε άλλος πωλητής με ανοιχτό κώδικα αποκαλύπτει μέρος, ή ολόκληρο τμήμα λογισμικού του προϊόντος τους, το κάνουν με την προσδοκία ότι οι δυνητικοί πελάτες που προσελκύονται από το ελεύθερο λογισμικό θα αγοράσουν στη συνέχεια συμβόλαια υποστήριξης, υπηρεσίες κατάρτισης και άλλες υπηρεσίες προστιθέμενης αξίας. Το άνοιγμα του λογισμικού τείνει να προσελκύει μέλη μιας κοινότητας που ποτέ δεν ήταν πιθανό να αγοράσουν το λογισμικό, ή συναφείς υπηρεσίες όπως είναι οι ανεξάρτητοι προγραμματιστές και οι εταιρείες που είναι πολύ μικρές - εκείνη τη στιγμή τουλάχιστον - για να παρέχουν λύσεις σε επίπεδο επιχείρησης. Έτσι, υπάρχουν σοβαροί λόγοι για το μεγάλο μέρος της συνεχιζόμενης δημιουργίας λύσεων ανοιχτού κώδικα από τους προμηθευτές λογισμικού επιχειρήσεων.

Είναι σαφές ότι είναι καιρός για ένα νέο οικονομικό μοντέλο. Οι προγραμματιστές που ενεργούν ως άτομα στην αγορά εργασίας θα μπορούσαν να αναμένουν κάποια

λογική πληρωμή για το χρόνο και την προσπάθεια που καταβάλλουν για την ανάπτυξη του OSS. Υπάρχουν πολλά πλεονεκτήματα του Ελεύθερου Λογισμικού για μη προγραμματιστές, όπως η ελευθερία επιλογής, ο έλεγχος από τον ιστό, η τιμή (δωρεάν), τα ανοιχτά πρότυπα και πολλά άλλα. Ωστόσο, παρόλο που τα οφέλη αυτά ισχύουν και για τους προγραμματιστές, υπάρχουν κάποιες δυσκολίες για τους προγραμματιστές, όπως η αδυναμία πληρωμής για εργασία (η αμοιβή είναι μηδενική ή ελάχιστη), καθιστώντας πιο δύσκολο για τους εμπορικούς φορείς να κερδίσουν χρήματα, μειώνοντας τις διαθέσιμες θέσεις εργασίας για προγραμματιστές που πληρώνονται (οι προγραμματιστές ανοιχτού κώδικα προσφέρουν το χρόνο τους). Η πλειοψηφία των προγραμματιστών λογισμικού που γράφουν κώδικα ανοιχτού τύπου το κάνουν ως μέρος των εργασιών τους. Το Apache, γράφτηκε αρχικά και εξακολουθεί να διατηρείται κυρίως από διαχειριστές δικτύων και προγραμματιστές που χρειάζονται αξιόπιστο, χαμηλού κόστους λογισμικό διακομιστή ιστού και πιστεύουν ότι είναι καλύτερο να συγκεντρώσουν τις προσπάθειές πάνω σε αυτό. Ο Firefox για παράδειγμα, ένας περιηγητής ιστού ανοιχτού κώδικα, είναι ένα από τα πιο χρησιμοποιημένα λογισμικά δωρεάν λογισμικού στην ιστορία των υπολογιστών. Το Mozilla Foundation, ιδιοκτήτης του Firefox, βγάζει τα περισσότερα από τα χρήματά του από την Google για παραπομπές από την αρχική σελίδα του Firefox και από το ενσωματωμένο πλαίσιο αναζήτησης.

Οι εταιρείες και οι οργανισμοί που ασχολούνται με την τεχνολογία της μηχανικής χρησιμοποίησαν στρατεύματα επαγγελματιών σχεδιαστών των οποίων ο πρωταρχικός ρόλος ήταν να λαμβάνουν σχέδια από τους μηχανικούς και τους τεχνικούς και να τα χρησιμοποιούν για την παραγωγή λεπτομερών και ευανάγνωστων τεχνικών σχεδίων που θα μπορούσαν να χρησιμοποιηθούν για τη δημιουργία ενός προϊόντος χωρίς την

άμεση επίβλεψη του σχεδιαστή. Πριν από την εμφάνιση του CAD και ιδιαίτερα πριν οι υπολογιστές καθίσταντο καθιερωμένοι, οι μηχανικοί δεν αναμενόταν να είναι σε θέση να κάνουν όλη τη διαδικασία σύνταξης μόνοι τους - παρόλο που θα έπρεπε να είχαν τη δυνατότητα θεωρητικά. Μετά το λογισμικό CAD εμφανίστηκε και αυτό το γεγονός άρχισε να αλλάζει. Το CAD όχι μόνο βοήθησε τους εργαζόμενους να δημιουργήσουν τρισδιάστατα μοντέλα των σχεδίων τους αλλά και να έρθουν συσκευασμένα με προγράμματα που παράγουν αυτόματα τεχνικά σχέδια με βάση αυτά τα μοντέλα 3D. Παρά τις παραπάνω έννοιες, σίγουρα θα υπάρχει πάντοτε η ανάγκη για την ανθρώπινη πινελιά και την κριτική συνεισφορά.

4. Βιβλιογραφία

1. 3D Innovations. (2013). The History of Computer-Aided Design (CAD) - 3D Innovations. [online] Available at: https://3d-innovations.com/blog/the-history-ofcomputer-aided-design-cad/ [Accessed 4 Apr. 2018].
2. Alexander, Christopher. 1965. "The Question of Computers in Design." Landscape.
3. Amidon, Jane. 2012. "Two Shifts and Four Threads." Topos. Amoroso, Nadia. 2012. Digital Landscape Architecture Now. New York: Thames & Hudson Inc.
4. Babar, A. (2016). HOW 3D MODELING MADE THINGS SMOOTH IN THE ENGINEERING?. [online] Indovance.com. Available at: http://www.indovance.com/how-3d-modeling-made-things-smooth-in-theengineering/ [Accessed 26 Mar. 2018].
5. Ballantyne, Andrew. 2007. Deleuze & Guattari for Architects. Milton Park: Routledge.
6. Coppinger, J. (2017). Review of FreeCAD 3D Modeling Software. [online] ThoughtCo. Available at: https://www.thoughtco.com/freecad-485328 [Accessed 4 Apr. 2018].
7. Freecadweb.org. (n.d.). About FreeCAD - FreeCAD Documentation. [online] Available at: https://www.freecadweb.org/wiki/About_FreeCAD [Accessed 7 Apr. 2018].
8. Freecadweb.org. (n.d.). Feature list - FreeCAD Documentation. [online] Available at: https://www.freecadweb.org/wiki/Feature_list [Accessed 19 Apr. 2018].

9. Freecadweb.org. (n.d.). Part Sweep - FreeCAD Documentation. [online] Available at: https://www.freecadweb.org/wiki/Part_Sweep [Accessed 14 Apr. 2018].
10. Help.solidworks.com. (n.d.). 2016 SOLIDWORKS Help - User Interface Overview. [online] Available at: http://help.solidworks.com/2016/English/SolidWorks/sldworks/c_user_interface_ overview.htm [Accessed 3 May 2018].
11. Hendrix, John Shannon. 2013. The Contradiction Between Form and Function in Architecture. Milton Park: Routledge.
12. Herlitz, Jeff. 2012. "No Scripting Required." Landscape Architecture Magazine. May.
13. Indovance.com. (2015). THE IMPORTANCE OF TECHNICAL DRAWINGS. [online] Available at: http://www.indovance.com/importance-technical-drawings/ [Accessed 26 Apr. 2018].
14. Jones, Stuart Mackenzie. 2008. "Visualizing Landscape Systems with Parametric Modeling." Athens: Williams College.
15. Kolarevic, Branko. 2003. Architecture in the Digital Age: Design and Manufacturing. New York: Spon Press.
16. ShoutMeTutorials.com. (2015). What is Solidworks CAD Software?. [online] Available at: http://shoutmetutorials.com/solidworks-basics/ [Accessed 12 Apr. 2018].
17. Steves-digicams.com. (n.d.). 6 Industries that Use 3D Modeling Software - Steve's Digicams. [online] Available at: http://www.stevesdigicams.com/knowledge-center/how-tos/video-software/6-industries-that-use3d-modeling-software.html#b [Accessed 6 Apr. 2018].

18. Spring. Allen, Stan. 2012. The Future That Is Now. March 12. http://places.designobserver.com/feature/architecture-school-the-futurethat-is-now/32728/. American Institute of Architects, California Council. 2012. Parametric Design: A Brief History. June 25. http://www.aiacc.org/2012/06/25/parametricdesign-a-brief-history/.
19. Types of mates. (2017).[ebook] p.1. Available at: http://www.webpages.uidaho.edu/mindworks/Solidworks/Topic%20References/ Types%20of%20Mates.pdf [Accessed 4 May 2018].