

Μηχανική Λογισμικού με Ανοιχτό Λογισμικό

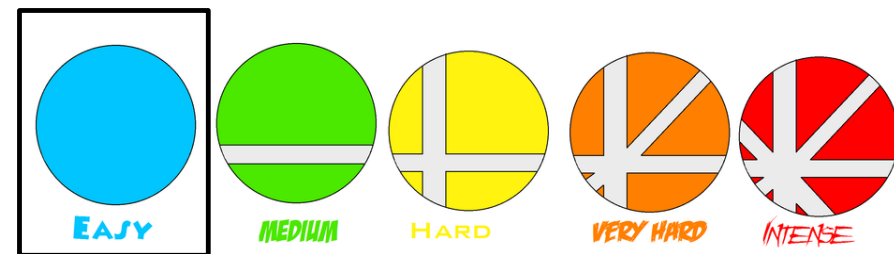
Δρ. Γεώργιος Κακαρόντζας

Τμήμα Μηχανικών Πληροφορικής Τ.Ε. – Α.Τ.Ε.Ι. Θεσσαλίας

The image shows a promotional poster for an event. At the top right is the logo of TEE (Κεντρικής & Δυτικής Θεσσαλίας). The main text in the center reads: 'ΕΠΙΣΤΗΜΟΝΙΚΗ ΗΜΕΡΙΔΑ', 'Ελεύθερο και ανοιχτό λογισμικό για Μηχανικούς στο Δημόσιο και Ιδιωτικό Τομέα'. Below this, the date and time are listed as '29.03.2018 | 18:00'. The location is 'ΛΑΡΙΣΑ | ΚΑΛΛΙΘΕΑΣ 7'. At the bottom, it says 'ΥΠΟΣΤΗΡΙΞΗ' followed by 'ΜΟΝΙΜΗ ΕΠΙΤΡΟΠΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ' and 'ΜΟΝΙΜΗ ΕΠΙΤΡΟΠΗ ΠΑΙΔΕΙΑΣ'. The background of the poster features faint code snippets and a circular graphic element.

Ατζέντα

- Εισαγωγή
- Εργαλεία Ανοιχτού Λογισμικού για Μηχανικούς Λογισμικού
- Χρήση και Διάθεση Ανοιχτού Λογισμικού για/από Μηχανικούς Λογισμικού
- Συμπεράσματα



Εισαγωγή

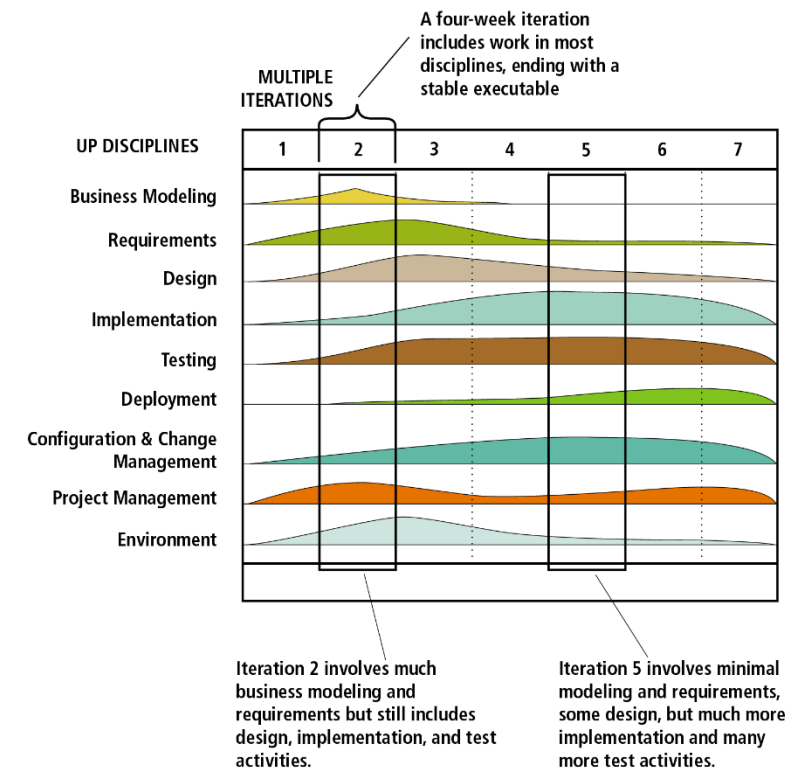
- Το ανοιχτό λογισμικού για τους μηχανικούς λογισμικού παρέχει τα ακόλουθα πλεονεκτήματα:
 - Αποτελεί μία οικονομική εναλλακτική λύση για την ανάπτυξη λογισμικού (π.χ. δωρεάν περιβάλλοντα ανάπτυξης λογισμικού).
 - Δίνει την δυνατότητα επαναχρησιμοποίησης βιβλιοθηκών και πλαισίων ανάπτυξης που κάνουν εφικτή την δημιουργία μεγάλων και χρήσιμων εφαρμογών από σχετικά μικρές ομάδες μηχανικών λογισμικού.
 - Δίνει πρόσβαση στον κώδικα για τροποποίηση και επαναχρησιμοποίηση (κάτω από προϋποθέσεις).
 - Αποτελεί πηγή γνώσης και μάθησης (π.χ. Ανάγνωση Κώδικα).
- Αλλά έχει και κάποια σημεία που χρήζουν προσοχής:
 - Η άδεια θα πρέπει να είναι συμβατή με τους σκοπούς διάθεσης του προϊόντος (για την περίπτωση του εμπορικού λογισμικού κυρίως).
 - Θα πρέπει να υπάρξει διασφάλιση της ποιότητας και της μακρόχρονης υποστήριξης.



Εργαλεία Ανοιχτού Λογισμικού για Μηχανικούς Λογισμικού

Τύποι εργαλείων

- Η μηχανική λογισμικού είναι μια διαδικασία που, μεταξύ άλλων, έχει τις εξής βασικές δραστηριότητες:
 - *Ανάλυση*: προσπάθεια αποτύπωσης των απαιτήσεων των χρηστών της υπό ανάπτυξης εφαρμογής.
 - *Σχεδίαση*: αποτύπωση μιας λύσης που ικανοποιεί τις απαιτήσεις.
 - *Υλοποίηση*: ανάπτυξη του κώδικα σύμφωνα με την σχεδίαση.
 - *Έλεγχος*: διασφάλιση της λειτουργικής ορθότητας της εφαρμογής και των ποιοτικών της χαρακτηριστικών.
- Οι πιο πάνω δραστηριότητες συνήθως αναμειγνύονται με διαφορετική ποσόστωση και ένταση κατά τη διάρκεια του έργου το οποίο αναπτύσσεται συνήθως σε σταθερού χρονικού μεγέθους επαναλήψεις (π.χ. μηνιαίες επαναλήψεις όλων των παραπάνω).
- Σε κάθε μία από τις πιο πάνω δραστηριότητες μπορούν να χρησιμοποιηθούν διαφορετικά εργαλεία ανοιχτού λογισμικού.



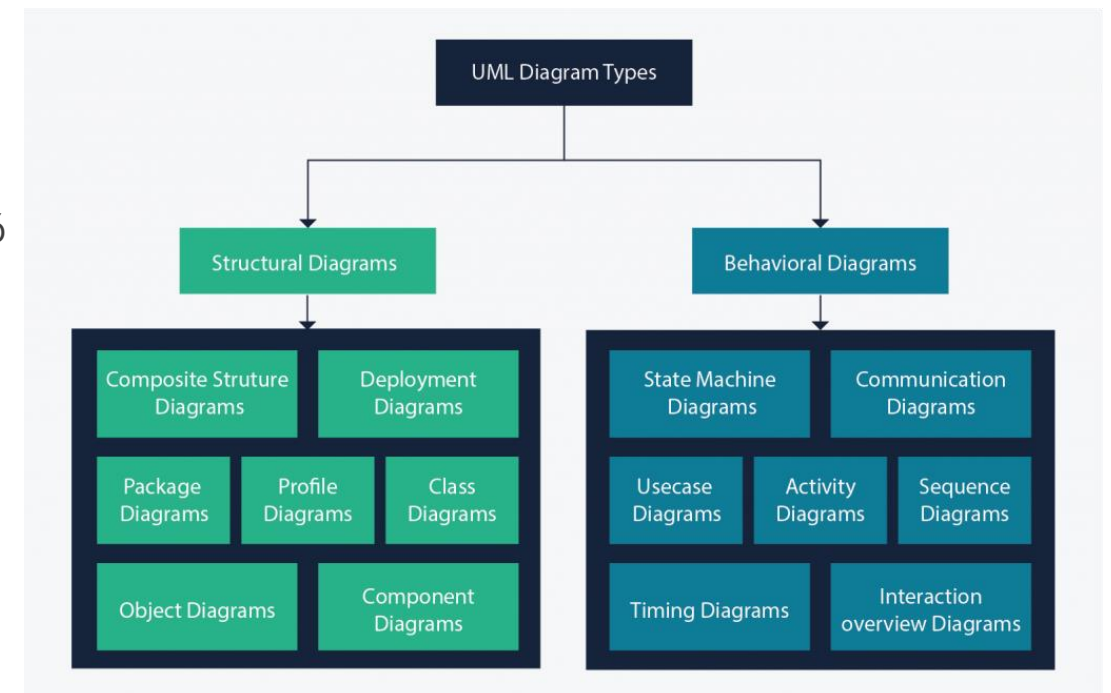
Ανάλυση Απαιτήσεων

- Συνήθως δεν χρησιμοποιούνται εξειδικευμένα εργαλεία. Οι απαιτήσεις σημειώνονται σε ένα κείμενο απαιτήσεων σε συμφωνία με τους πελάτες και χρησιμοποιούνται κάποια πρότυπα έγγραφα (πίνακες απαιτήσεων).
- Ενδεχομένως να χρησιμοποιηθούν και κάποια εργαλεία με τα οποία γίνονται διαγράμματα περιπτώσεων χρήσης.
- Πολλές φορές απλώς οι απαιτήσεις καταγράφονται σε κάποιο Wiki (μια διαδικτυακή εφαρμογή, η οποία επιτρέπει στους χρήστες της να επεξεργάζονται το περιεχόμενό της σε συνεργασία με τους άλλους).
- Σχετικά Open Source εργαλεία:
 - Libre Office (καταγραφή απαιτήσεων με κείμενο) - <https://el.libreoffice.org/>
 - Umbrello, Papyrus κ.α. (διαγράμματα περιπτώσεων χρήσης) - <https://umbrello.kde.org/>, <https://eclipse.org/papyrus/>
 - Wiki (συνεργατική online επεξεργασία) – Λύσεις που παρέχονται από το σύστημα ελέγχου εκδόσεων λογισμικού (π.χ. GitHub) ή εσωτερικές λύσεις σε εταιρίες (π.χ. MediaWiki - <https://www.mediawiki.org/wiki/MediaWiki>)

Τμήμα Περίπτωσης Χρήσης	Σχόλιο
Όνομα Περίπτωσης Χρήσης	Ξεκινήστε με ρήμα
Εμβέλεια	Το υπό σχεδίαση σύστημα
Επίπεδο	«στόχος χρήστη» ή «υπολειτουργία»
Πρωτεύων Χειριστής	Καλεί το σύστημα για την διενέργεια των υπηρεσιών του.
Ενδιαφερόμενοι και ενδιαφέροντα	Ποιοι ενδιαφέρονται για την περίπτωση χρήσης και τι θέλουν να πετύχουν.
Προσυνθήκες	Τι θα πρέπει να είναι αληθές κατά την έναρξη και αξίζει να γνωρίζει ο αναγνώστης.
Εγγυήσεις Επιτυχίας	Τι θα πρέπει να είναι αληθές κατά την επιτυχημένη ολοκλήρωση και αξίζει να γνωρίζει ο αναγνώστης.
Βασική ακολουθία ενεργειών / αποκρίσεων	Ένα τυπικό, άνευ συνθήκης, σενάριο επιτυχίας (η λεγόμενη «ευτυχής διαδρομή»).
Εναλλακτικές ακολουθίες ενεργειών / αποκρίσεων	Εναλλακτικά σενάρια επιτυχίας και αποτυχίας.

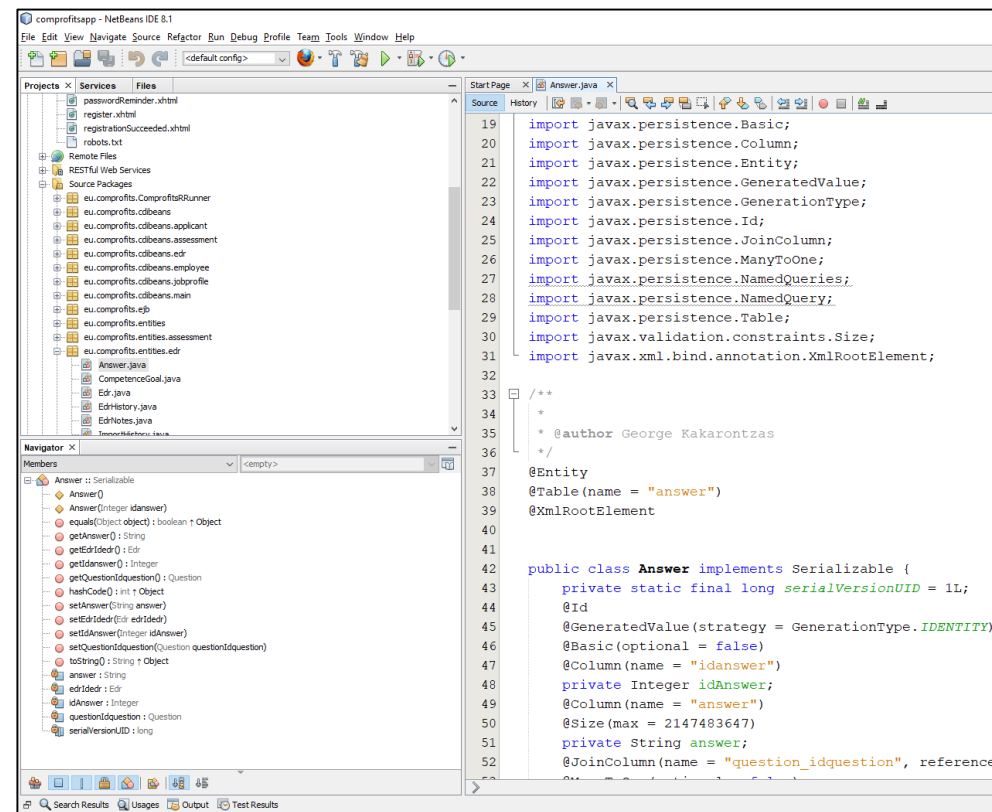
Σχεδίαση Λογισμικού

- Η σχεδίαση του λογισμικού γίνεται συνήθως σε μία γλώσσα που ονομάζεται **UML (Unified Modeling Language)** και δίνει την δυνατότητα στους μηχανικούς λογισμικού να αποτυπώσουν το σχέδιο του λογισμικού με διαγράμματα (κλάσεων, ακολουθίας, συστατικών κ.α.)
- Συνήθως χρειάζονται πολλά διαφορετικά διαγράμματα γιατί το υπό ανάπτυξη λογισμικό έχει διαφορετικές απόψεις που θα πρέπει να σχεδιαστούν (στατική, δυναμική, αρχιτεκτονική, περιπτώσεων χρήσης κ.α.)
- Κυριότερα προγράμματα ανοιχτού λογισμικού για σχεδίαση με UML:
 - Umbrello: <https://umbrello.kde.org/>
 - Papyrus: <https://eclipse.org/papyrus/>
 - ArgoUML: <http://argouml.tigris.org/>
 - Open ModelSphere: <http://www.modelsphere.com/org/>



Υλοποίηση Λογισμικού

- Η υλοποίηση λογισμικού γίνεται συνήθως με Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης (Integrated Development Environments – IDEs).
- Αρκετά δημοφιλή περιβάλλοντα ανάπτυξης λογισμικού είναι ανοιχτό λογισμικό:
 - Eclipse IDE: <http://www.eclipse.org/>
 - NetBeans IDE: <https://netbeans.org/>
 - Android Studio: <https://developer.android.com/studio/index.html>, κ.α.
- Τα IDEs υποστηρίζουν συνήθως όλες τις διαδικασίες ανάπτυξης λογισμικού που αφορούν την κωδικοποίηση, τον έλεγχο και τον διαμοιρασμό του κώδικα μέσω αποθετηρίων (π.χ. GitHub).
- Υποστηρίζουν τον μηχανικό λογισμικού με πολλούς διαφορετικούς τρόπους (αυτόματη συμπλήρωση κώδικα, παραγωγή κώδικα, γραφική δημιουργία διασυνδέσεων χρήστη και πολλά άλλα).

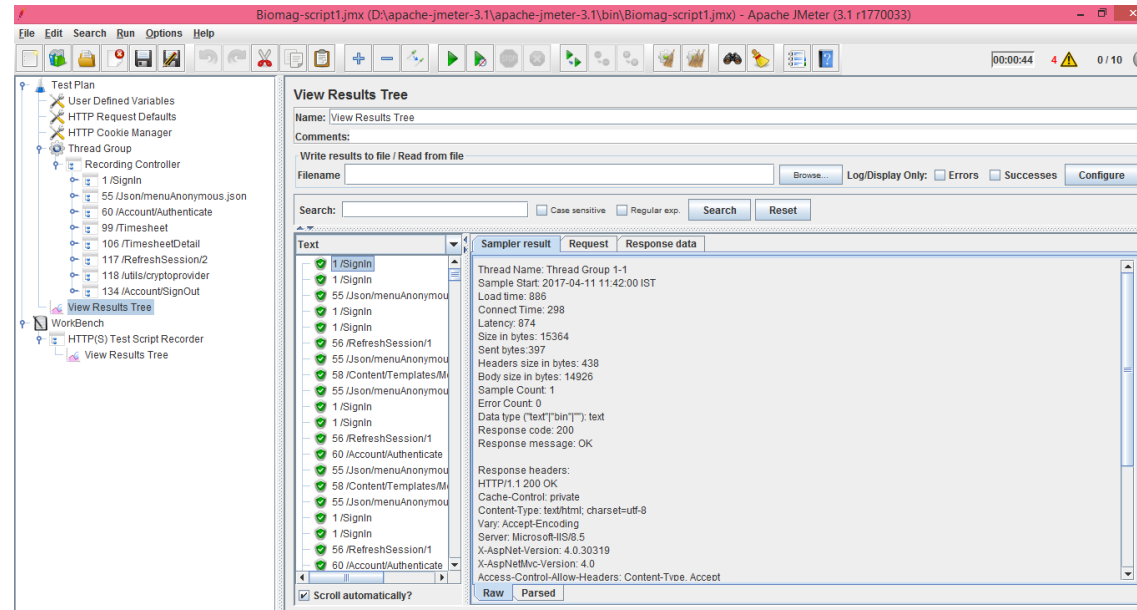


The screenshot shows the NetBeans IDE interface. The left sidebar displays a project tree for 'comprofitsapp' with various source files and packages. The main editor window shows the 'Answer.java' file with the following code:

```
19 import javax.persistence.Basic;
20 import javax.persistence.Column;
21 import javax.persistence.Entity;
22 import javax.persistence.GeneratedValue;
23 import javax.persistence.GenerationType;
24 import javax.persistence.Id;
25 import javax.persistence.JoinColumn;
26 import javax.persistence.ManyToOne;
27 import javax.persistence.NamedQueries;
28 import javax.persistence.NamedQuery;
29 import javax.persistence.Table;
30 import javax.validation.constraints.Size;
31 import javax.xml.bind.annotation.XmlRootElement;
32
33 /**
34  *
35  * @author George Kakarontzas
36  */
37 @Entity
38 @Table(name = "answer")
39 @XmlRootElement
40
41
42 public class Answer implements Serializable {
43     private static final long serialVersionUID = 1L;
44     @Id
45     @GeneratedValue(strategy = GenerationType.IDENTITY)
46     @Basic(optional = false)
47     @Column(name = "idanswer")
48     private Integer idAnswer;
49     @Column(name = "answer")
50     @Size(max = 2147483647)
51     private String answer;
52     @JoinColumn(name = "question_idquestion", referenced
```


Έλεγχος λογισμικού

- Ο έλεγχος λογισμικού διασφαλίζει την ορθότητα και την ποιότητα του λογισμικού:
 - *Λειτουργική ορθότητα:* Κατά την ανάπτυξη λογισμικού γίνονται έλεγχοι από τους ίδιους τους μηχανικούς λογισμικού, οι οποίοι έχουν στόχο των έλεγχου μεμονωμένων μονάδων του προγράμματος που αναπτύσσεται. Οι έλεγχοι αυτοί έχουν ως στόχο την εύρεση λαθών και την διόρθωσή τους πριν το προϊόν του λογισμικού διατεθεί στους τελικούς χρήστες.
 - *Ποιοτικός έλεγχος:* Αφορά άλλα ποιοτικά χαρακτηριστικά που μπορεί να είναι σημαντικά για μία εφαρμογή (π.χ. ταχύτητα-απόδοση, ασφάλεια, ευχρηστία κ.α.) και έχουν στόχο την εύρεση προβλημάτων σχετικά με αυτά τα ποιοτικά χαρακτηριστικά και την διόρθωσή τους πριν το προϊόν του λογισμικού διατεθεί στους τελικούς χρήστες.
- Εργαλεία ανοιχτού λογισμικού σχετικά με τον έλεγχο λογισμικού:
 - JUnit – <http://junit.org>
 - NUnit – <http://nunit.org>
 - Apache JMeter - <http://jmeter.apache.org/>

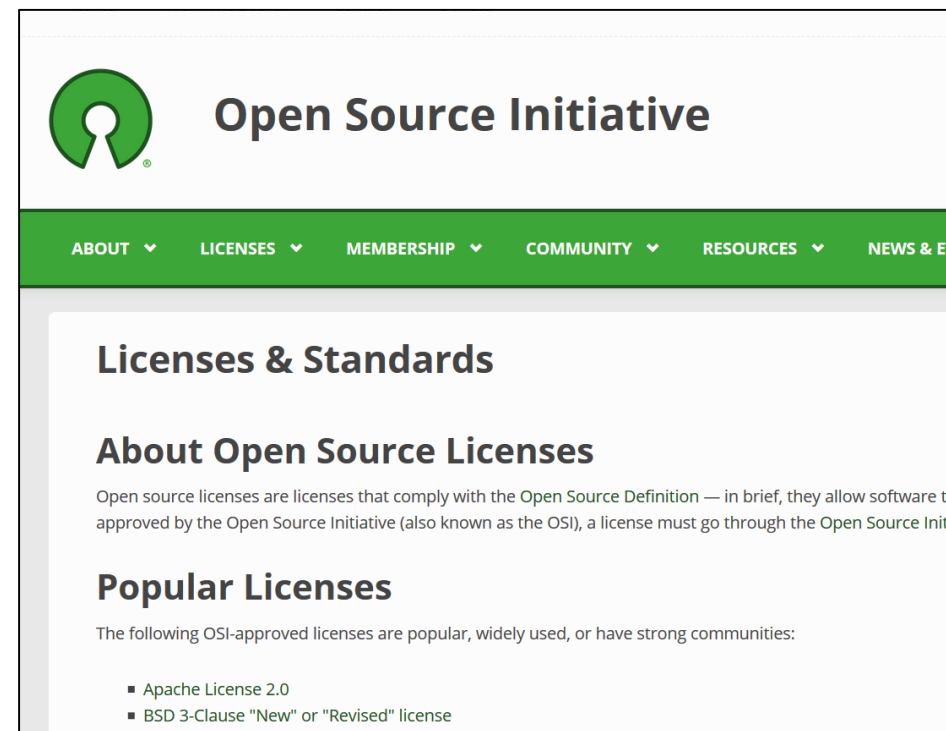




Χρήση και Διάθεση Ανοιχτού Λογισμικού για Μηχανικούς Λογισμικού

Άδειες Ανοιχτού Λογισμικού

- Οι άδειες ανοιχτού λογισμικού δεν περιορίζουν την χρήση λογισμικού αλλά ενδέχεται να θέσουν περιορισμούς στην διάθεση του ανοιχτού λογισμικού και μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες:
 - Άδειες copyleft (όρος που υποδηλώνει την αντιδιαστολή με το copyright): αυτές θέτουν περιορισμούς για εμπορική χρήση.
 - Άδειες επιτρεπτικές: δεν θέτουν ουσιαστικούς περιορισμούς για εμπορική χρήση.
- Οι άδειες λογισμικού επηρεάζουν τους μηχανικούς λογισμικού με δύο τρόπους:
 - Όταν χρησιμοποιούν ανοιχτό λογισμικό στα δικά τους προϊόντα θα πρέπει να είναι συμβατή η άδεια του ανοιχτού λογισμικού με την άδεια του δικού τους λογισμικού.
 - Όταν διαθέτουν δικά τους προγράμματα ως ανοιχτό λογισμικό θα πρέπει να επιλέξουν την κατάλληλη άδεια ανάλογα με τους στόχους που έχουν.



The screenshot shows the Open Source Initiative (OSI) website. The header features the OSI logo (a green circle with a white 'O' shape) and the text "Open Source Initiative". Below the header is a green navigation bar with links for "ABOUT", "LICENSES", "MEMBERSHIP", "COMMUNITY", "RESOURCES", and "NEWS & EVENTS". The main content area is titled "Licenses & Standards" and includes a sub-section "About Open Source Licenses" with a brief definition of open source licenses. Below this is a section titled "Popular Licenses" which lists "Apache License 2.0" and "BSD 3-Clause 'New' or 'Revised' license".

Μερικές τυπικές άδειες

Άδεια	Σημασία
GNU GPLv3	Ισχυρή copyleft άδεια. Απαιτεί πλήρη διάθεση του πηγαίου κώδικα και των τροποποιήσεων αυτού από άλλους κάτω από την ίδια άδεια. Οι σημειώσεις της άδειας θα πρέπει να διατηρηθούν. Υπάρχει ρητή παραχώρηση δικαιωμάτων ευρεσιτεχνίας (express grant of patent rights).
Apache License 2.0	Μία επιτρεπτική άδεια. Απαιτεί κυρίως διατήρηση των σημειώσεων της άδειας. Επίσης παρέχει ρητή παραχώρηση δικαιωμάτων ευρεσιτεχνίας. Μεγαλύτερα προγράμματα που χρησιμοποιούν λογισμικό υπό αυτήν την άδεια μπορούν να διατεθούν με διαφορετική άδεια και χωρίς να υπάρχει υποχρέωση ανοιχτού κώδικα.
MIT License	Απλούστερη επιτρεπτική άδεια με απαιτήσεις μόνο σε ότι αφορά της σημειώσεις της άδειας. Μεγαλύτερα προγράμματα που χρησιμοποιούν λογισμικό υπό αυτήν την άδεια μπορούν να διατεθούν με διαφορετική άδεια και χωρίς να υπάρχει υποχρέωση ανοιχτού κώδικα.

Κριτήρια Επιλογής Ανοιχτού Λογισμικού από Μηχανικούς Λογισμικού

- Φήμη του ανοιχτού λογισμικού: είναι το ανοιχτό λογισμικό δημοφιλές; Χρησιμοποιείται ευρέως;
- Κοινότητα υποστήριξης: υπάρχει ενεργή ανάπτυξη και υποστήριξη από μία μεγάλη κοινότητα μηχανικών λογισμικού;
- Πρότυπα και Διαλειτουργικότητα: είναι συμβατό με σχετικά πρότυπα;
- Ύπαρξη εμπορικής υποστήριξης: υπάρχει δυνατότητα επί πληρωμής υποστήριξης για το λογισμικό;
- Τεκμηρίωση: υπάρχει καλή τεκμηρίωση; (βιβλία, websites κ.λπ.)
- Άδεια: είναι η άδεια του λογισμικού συμβατή με την άδεια του υπό ανάπτυξη λογισμικού;

Κριτήρια επιλογής άδειας ανοιχτού λογισμικού

- Σε ποιον απευθύνεται το λογισμικό (επιχειρήσεις, φοιτητές, ακαδημαϊκούς, απλούς χρήστες); Ανάλογα με την κατηγορία πρέπει να επιλεγεί και η άδεια. Για παράδειγμα για τις περισσότερες επιχειρήσεις παραγωγής λογισμικού μία άδεια copyleft θα ήταν απαγορευτική, έτσι να στοχεύουμε ένα λογισμικό που παράγουμε να χρησιμοποιηθεί σε εμπορικά προϊόντα άλλων με στόχο κέρδη από εκπαίδευση κ.α. θα πρέπει το προϊόν να είναι με άδεια επιτρεπτική.
- Πόσοι και ποιοι αναμένεται να συνεισφέρουν στο ανοιχτό λογισμικό και ποια είναι τα δικά τους ενδιαφέροντα; Για παράδειγμα αν το έργο θέλει να προσελκύσει πολλούς ανεξάρτητους μηχανικούς λογισμικού απ' όλο το κόσμο, μία άδεια copyleft θα δουλέψει καλύτερα, γιατί εξασφαλίζει πως η δουλειά τους δεν θα γίνει προϊόν εκμετάλλευσης τρίτων.
- Ποιο είναι το επιχειρηματικό μοντέλο το οποίο θέλουμε να ακολουθήσουμε με το ανοιχτό λογισμικό που παράγουμε; Δηλ. πως θα χρηματοδοτηθεί αυτό το λογισμικό; Υπάρχουν πολλά επιχειρηματικά μοντέλα: dual-licensing, πώληση υπηρεσιών γύρω από το λογισμικό, πώληση άλλων εμπορευμάτων, πώληση του λογισμικού ως υπηρεσία κ.α. (https://en.wikipedia.org/wiki/Business_models_for_open-source_software)

Συμπεράσματα

- Το ανοιχτό λογισμικό αποτελεί σήμερα την βασική εργαλειοθήκη του μηχανικού λογισμικού.
- Η δυνατότητα επαναχρησιμοποίησης ανοιχτού λογισμικού από μηχανικούς λογισμικού καθιστά εφικτή την δημιουργία μεγάλων και αξιοσημείωτων εφαρμογών από σχετικά μικρές ομάδες και σε σχετικά μικρό χρονικό διάστημα.
- Απαιτείται προσοχή στην άδεια του λογισμικού που ενσωματώνεται για την παραγωγή νέου λογισμικού, έτσι ώστε αυτή να είναι συμβατή με τις προθέσεις της εταιρίας που αναπτύσσει το νέο λογισμικό, ιδιαίτερα αν η άδεια είναι copyleft.
- Απαιτείται προσοχή στην επιλεγμένη άδεια ενός νέου ανοιχτού λογισμικού, προκειμένου να είναι δυνατή η χρηματοδότησή του ανάλογα και με το επιλεγμένο επιχειρηματικό μοντέλο και το κοινό στο οποίο απευθύνεται η νέα εφαρμογή.

Ερωτήσεις;



Κατεβάστε αυτή την παρουσίαση από την διεύθυνση:
<https://goo.gl/mGVVei>