

Περιεχόμενα

Κατάλογος Σχημάτων

iii

1	Εισαγωγή	1
1.1	Σημασία του προβλήματος	1
1.2	Στόχοι της εργασίας	1
1.3	Μεθοδολογία προσέγγισης	1
1.4	Διάρθρωση της διπλωματικής εργασίας	1
2	Διαχείριση Εργασιών	2
2.1	Το πρόβλημα της διαχείρισης εργασιών	2
2.1.1	Εργασία	2
2.1.2	Ορισμός της διαχείρισης εργασιών	2
2.1.3	Πεδίο εφαρμογής διαχείρισης εργασιών	2
2.1.4	Διαφορά με διαχείριση έργου	3
2.2	Ιστορική αναδρομή	3
2.2.1	Προφορικότητα και μνημονικές συσκευές	4
2.2.2	Πρώτα ημερολόγια	5
2.2.3	Σύγχρονη εποχή	5
2.3	Η συνδρομή της τεχνολογίας	6
2.3.1	Ψηφιακά εργαλεία	6
2.3.1.1	Microsoft Project	6
2.4	Μεθοδολογίες	7
2.4.1	Διάγραμμα Γκαντ	7
2.4.2	Program evaluation and review technique (PERT)	8
2.4.3	Μέθοδος κρίσιμης διαδρομής (Critical Path Method – CPM)	8
2.4.4	Agile και Kanban	8
2.5	Διαχείριση έργων στο πανεπιστήμιο	8
2.5.1	Προβλήματα διαχείρισης που αντιμετωπίζουν οι φοιτητές	8
2.5.2	Χαρακτηριστικά που οι φοιτητές θα επιθυμούσαν σε μια εφαρμογή	9

3	Low-Code	10
3.1	Ορισμός	10
3.2	Πριν το Low-Code	11
3.2.1	Computer-Aided Software Enginnering (CASE)	11
3.2.2	Model-driven Architecture	12

Κατάλογος Σχημάτων

2.1	Η συσκευή λουκάσα	4
2.2	Η συσκευή κουίπου	5
2.3	Στιγμιότυπο από το Microsoft Project 3.0 (σε DOS) [20]	6
2.4	Στιγμιότυπο από το Microsoft Project 2000 [20]	7

Κεφάλαιο 1

Low-Code

Τα πετρογραφικά και οι ζωγραφιές στους τοίχους παλαιών σπηλαίων είναι ένα δείγμα από το πόσο ουσιώδης είναι η οπτική επικοινωνία για τον άνθρωπο. [9]

1.1 Ορισμός

Μια Πλατφόρμα Ανάπτυξης Εφαρμογών σε Low-Code (LCAP) είναι μια πλατφόρμα ανάπτυξης λογισμικού που υποστηρίζει την ταχεία ανάπτυξη και διαχείριση εφαρμογών. Συνήθως είναι Platform-as-a-service (PaaS) cloud μοντέλα, και χρησιμοποιείται ελάχιστος ή και μηδενικός δομημένος προγραμματισμός (structured programming).

Για τον προγραμματισμό παρέχεται γραφικό περιβάλλον με οπτικές αφαιρέσεις (visual abstractions), μάλιστα επιτρέποντας χρήστες χωρίς προγραμματιστική εμπειρία να συνεισφέρουν στην ανάπτυξη του λογισμικού χωρίς είναι αναγκαία η βοήθεια των προγραμματιστών. Έτσι οι προγραμματιστές εστιάζουν παραπάνω στη σχεδίαση της εφαρμογής, χωρίς να ξοδεύουν χρόνο άσκοπα σε λεπτομέρειες.

Με λίγα λόγια, σκοπός είναι η παραγωγική ανάπτυξη λογισμικού με τη λιγότερη δυνατή προσπάθεια και με χαμηλότερο κόστος, και η εύκολη προσαρμογή του λογισμικού στις ταχέως μεταβαλλόμενες συνθήκες των σημερινών λειτουργικών συστημάτων. Η χρήση των LCAP έχει τύχει θετικής αποδοχής από τη βιομηχανία και η υιοθέτησή τους αυξάνεται συνεχώς. [1, 2, 16]

“When you can visually create new business applications with minimal hand-coding –when your developers can do more of greater value, faster– that’s low-code.” [19]

1.2 Πριν το Low-Code

Φυσικά η μηχανική λογισμικού¹ έχει περάσει πολλά στάδια στην ιστορία της μέχρι να αρχίσουμε να αναφερόμαστε και σε χρήση αφαιρέσεων. Μέχρι τη δεκαετία του 1970, η ανάπτυξη πληροφοριακών συστημάτων έμοιαζε παραπάνω ως μια τέχνη παρά ως επιστήμη, καθώς δεν ακολουθούσε κάποια δόμηση. Αντιθέτως, ο προγραμματιστής λάμβανε μια σειρά από απαιτήσεις και ανάγκες από την πλευρά του χρήστη, και μετά από ένα διάστημα παρέδιδε ένα σύστημα που συνήθως δεν κάλυπτε εξ' ολοκλήρου όλες τις απαιτήσεις του χρήστη αλλά ήταν σίγουρα καλύτερο από το τίποτα. Η συγκεκριμένη μέθοδος, αποκαλούμενη ως **κλασική μέθοδος**, χαρακτηρίζεται από ανεπίσημες οδηγίες, έλλειψη τυποποίησης και αναφορών (documentation).

Η ανάγκη για αύξηση της παραγωγικότητας στον κύκλο ζωής έκδοσης λογισμικού (software release life cycle)² οδήγησε στη δημιουργία **επίσημων μεθόδων** στα τέλη της δεκαετίας του 1970. Σκοπός τους ήταν η τυποποίηση του σχεδιασμού με στόχο τη βελτίωση της ποιότητάς του. Παράδειγμα αυτών των τεχνικών είναι η χρήση δομημένης ανάλυσης (structured analysis). Έτσι, οι μηχανικοί μπορούσαν να φτιάξουν διαγράμματα ροών δεδομένων (data flow diagrams)³, μοντέλα οντοτήτων-συσχετίσεων (ER – entity-relationship models)⁴, δημιουργώντας μια συστημική περιγραφή του λογικού και φυσικού μέρους του πληροφοριακού συστήματος που ανέπτυσαν.

Με την περαιτέρω ανάπτυξη των προσωπικών υπολογιστών στα μέσα της δεκαετίας του 1980, η χρήση επίσημων μεθόδων ήταν μονόδρομος, και μάλιστα οδήγησε στην άνθηση αυτόματων περιβαλλόντων και εργαλείων. Αυτά τα περιβάλλοντα, γνωστά ως CASE, επέτρεπαν τους μηχανικούς να καταγράφουν και να μοντελοποιούν με συστηματικό τρόπο ένα πληροφοριακό σύστημα από τις αρχικές περιγραφές του χρήστη ως τη σχεδίαση και την υλοποίηση και να εκτελούν δοκιμές για τη συνέπειά του. [5, 4]

1.2.1 Computer-Aided Software Engineering (CASE)

Μέχρι πρότινος, τα εργαλεία που αφορούν την ανάπτυξη λογισμικού εστιάζουν κυρίως στην επεξεργασία πηγαίου κώδικα και την αποσφαλμάτωση του. Σε αντίθεση λοιπόν με τα υπάρχοντα εργαλεία, τα CASE περιβάλλοντα βοηθούν στη μεθοδολογία της ανάπτυξης λογισμικού: στην ανάλυση απαιτήσεων, στο λογικό σχεδιασμό, στον έλεγχο εγκυρότητας, την επαναχρησιμοποίηση και εξάλειψη πλεονασμών.

¹Θα ορίζαμε τη μηχανική λογισμικού ως μια πειθαρχημένη και αυστηρή εφαρμογή μεθόδων, διαδικασιών και εργαλείων στη διαχείριση και ανάπτυξη υπολογιστικών συστημάτων. Πρόκειται για ένα εννοιολογικό πλαίσιο που περιγράφει τη διαχείριση συστημάτων.

²Πρόκειται μια έννοια που αναφέρεται στις φάσεις ανάπτυξης και ύπαρξης ενός λογισμικού. Ξεκινάει από τη σύλληψη της ιδέας, τη μελέτη για τις απαιτήσεις, την υλοποίησή του, τη διάθεση του προϊόντος στο πελάτη, τη υποστήριξή του με ενημερώσεις και τέλος την απόσυρσή του.

³Είναι μια οπτική αναπαράσταση της ροής των δεδομένων σε ένα σύστημα. Αναγράφονται οι διεργασίες (κύκλοι), οι είσοδοι και έξοδοι (τετράγωνα) και η αποθήκευση των δεδομένων (παράλληλες γραμμές).

⁴Περιγράφει ένα σύνολο αντικειμένων (οντότητες) και τις σχέσεις μεταξύ αυτών των αντικειμένων.

Είναι το δεξί χέρι ενός μηχανικού λογισμικού, βοηθώντας σε πολλές εργασίες που τον δυσκολεύουν, αυξάνοντας την παραγωγικότητα και την ποιότητα της δουλειάς του. Τα εργαλεία που περιλαμβάνουν ποικίλλουν: κάποια επιτρέπουν τη δημιουργία διαγράμμάτων ενώ άλλα μπορούν να αυτοματοποιούν όλα τα στάδια του κύκλου ζωής έκδοσης λογισμικού. Ένα ολοκληρωμένο CASE περιβάλλον περιλαμβάνει:

- ένα διαδραστικό, φιλικό για το χρήστη, οπτικό περιβάλλον διαχείρισης
- ένα σύνολο από εργαλεία ανάπτυξης (επεξεργαστές κειμένου, λεξικά, αναλυτές σχεδιασμού κ.α.)
- ένα σύνολο από εργαλεία για τον έλεγχο της διαδικασίας (για τον χρονοπρογραμματισμό, τη διασφάλιση ποιότητας κ.α.)
- ένα περιβάλλον βοήθειας με το documentation των εργαλείων
- ένα σύστημα διαχείρισης βάσεων δεδομένων

Τα συστήματα που δημιουργούνται από το CASE είναι εφαρμογές της πειθαρχημένης εφαρμογής μεθόδων της μηχανικής λογισμικού. Τα CASE περιβάλλοντα έθεσαν τα θεμέλια για τη δημιουργία νέων προτύπων, όπως ο οπτικός προγραμματισμός (visual programming) και ο προγραμματισμός που βασίζεται σε μοντέλα. [4, 9, 12]

1.2.2 Model-driven Architecture

Οι αρχιτεκτονικές που βασίζονται σε μοντέλα (model-driven architecture - MDA) διαχωρίζουν τη λειτουργικότητα μιας εφαρμογής από την υλοποίησή της σε μια συγκεκριμένη πλατφόρμα, προσφέροντας ένα υψηλότερο επίπεδο αφαίρεσης. Στόχος είναι οι προγραμματιστές να εστιάζουν περισσότερο στον σχεδιασμό και λιγότερο στο να λύνουν θέματα που αφορούν την πλατφόρμα υλοποίησης.

Μέχρι πρότινος η ανάπτυξη λογισμικού αφορούσε καθαρά δομικό κώδικα. Η MDA άλλαξε τον τρόπο σκέψης των προγραμματιστών, καθώς πλέον επικεντρώνονταν παρπάνω στον σωστό διαχωρισμό των χαρακτηριστικών, στην αφαιρετικότητα και στην αυτοματοποίηση. [2, 11]

Βιβλιογραφία

- [1] Alexander C. Bock και Ulrich Frank. «Low-Code Platform». Στο: *Business and Information Systems Engineering* 63 (6 Δεκ. 2021), σσ. 733–740. ISSN: 18670202. DOI: 10.1007/s12599-021-00726-8.
- [2] Alessio Bucaioni, Antonio Cicchetti και Federico Ciccozzi. «Modelling in low-code development: a multi-vocal systematic review». Στο: *Software and Systems Modeling* 21 (5 Οκτ. 2022), σσ. 1959–1981. ISSN: 16191374. DOI: 10.1007/s10270-021-00964-0.
- [3] *BUS402: History of Project Management* | Saylor Academy — learn.saylor.org. <https://learn.saylor.org/mod/page/view.php?id=65663>. [Accessed 26-10-2024].
- [4] Albert E Case. *Computer-aided software engineering (case): technology for improving software development productivity*. 1985.
- [5] E. J. Chikofsky. *Software Development — Computer-Aided Software Engineering (CASE)*.
- [6] Ryoko Fukuzawa, Hideo Joho και Tetsuya Maeshiro. «Practice and experience of task management of university students: Case of University of Tsukuba, Japan». Στο: *Education for Information* 31 (3 Ιούλ. 2015), σσ. 109–124. ISSN: 01678329. DOI: 10.3233/EFI-150953.
- [7] Jack Goody. «Memory in Oral Tradition». Στο: Cambridge University Press, Μαρ. 2013, σσ. 73–94. DOI: 10.1017/cbo9781139171137.005.
- [8] *Hoover Dam – the Greatest Project in Times of the Great Depression. What Can Be Done to Achieve Success? - Strefa PMI* — strefapmi.pl. <https://strefapmi.pl/strefa-studenta/hoover-dam-the-greatest-project-in-times-of-the-great-depression/>. [Accessed 30-10-2024].
- [9] D. L. Kuhn. *Selecting and Effectively Using a Computer-Aided Software Engineering Tool*. 1989.
- [10] *Manage Your Team’s Projects From Anywhere* | Trello — trello.com. <https://trello.com/>. [Accessed 16-10-2024].
- [11] *MDA FAQ* | Object Management Group — [omg.org](https://www.omg.org). https://www.omg.org/mda/faq_mda.htm. [Accessed 11-10-2024].
- [12] G. Premkumar και Michael Potter. *Adoption of Computer Aided Software Engineering (CASE) Technology: An Innovation Adoption Perspective*.

- [13] *Quipu* - Wikipedia — *en.wikipedia.org*. <https://en.wikipedia.org/wiki/Quipu>. [Accessed 22-10-2024].
- [14] Thomas Q Reefer. *Lukasa: A Luba Memory Device*. doi: doi:10.2307/3335144.
- [15] E. G. Richards. *Mapping time: The calendar and its history*. Oxford University Press, 2000.
- [16] Apurvanand Sahay κ.ά. «Supporting the understanding and comparison of low-code development platforms». Στο: *Proceedings - 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020*. Institute of Electrical και Electronics Engineers Inc., Αύγ. 2020, σσ. 171–178. ISBN: 9781728195322. doi: 10.1109/SEAA51224.2020.00036.
- [17] *Todoist | A To-Do List to Organize Your Work & Life* — *todoist.com*. <https://todoist.com/>. [Accessed 16-10-2024].
- [18] Julia Castillo Trujillo. *Designing A Time Management App For And With Informatics Students*. 2020.
- [19] *What Is Low-Code? | IBM* — *ibm.com*. <https://www.ibm.com/topics/low-code>. [Accessed 11-10-2024].
- [20] *WinWorld: Welcome* — *winworldpc.com*. <https://winworldpc.com/home>. [Accessed 31-10-2024].
- [21] *Your connected workspace for wiki, docs & projects | Notion* — *notion.so*. <https://www.notion.so/>. [Accessed 16-10-2024].
- [22] Μιχαήλ Ξένος. *Ποιότητα Λογισμικού*. GOTSIS, 2021. ISBN: 9786185560102.