

Περιεχόμενα

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	<Πρόβλημα διαχείρισης εργασιών>	2
2.2	Διαχείριση εργασιών στο πανεπιστήμιο	2
2.2.1	Προβλήματα διαχείρισης που αντιμετωπίζουν οι φοιτητές	2
2.2.2	Χαρακτηριστικά που οι φοιτητές θα επιθυμούσαν σε μια εφαρμογή	2
3	Low code	4
3.1	Ορισμός	4
3.2	Προηγούμενες προσπάθειες	5
3.2.1	Computer-Aided Software Enginnering	5
3.2.2	Model-driven Architecture	5
4	Mendix	6
5	<ανάπτυξη εφαρμογής>	7

Κεφάλαιο 1

Εισαγωγή

1.1 Σημασία του προβλήματος

1.2 Στόχοι της εργασίας

1.3 Μεθοδολογία προσέγγισης

1.4 Διάρθρωση της διπλωματικής εργασίας

Κεφάλαιο 2

Διαχείριση Εργασιών

2.1 <Πρόβλημα διαχείρισης εργασιών>

Studies of Task Management Towards the Design of a Personal Task List Manager
fds

2.2 Διαχείριση εργασιών στο πανεπιστήμιο

2.2.1 Προβλήματα διαχείρισης που αντιμετωπίζουν οι φοιτητές

Σε έρευνα [4] που διεξήχθη στο Πανεπιστήμιο του Τσουκούμπα της Ιαπωνίας η οποία διερευνούσε τη διαχείριση του προγραμματισμού των εργασιών των φοιτητών, παρατηρήθηκε πως η πλειοψηφία των φοιτητών αντιμετωπίζει δυσκολίες στην εκκίνηση μιας νέας εργασίας με βασικούς λόγους: α) την έλλειψη χρόνου (26,9%), β) την αγνόησή της επειδή τη θεωρούσαν ελάχιστον σημασίας (15,7%), γ) επειδή την ξέχασαν (12,3%), δ) λόγω κακής συνεργασίας (11,2%) και ε) επειδή ήταν κουραστική (8,9%). Παρατηρούμε πως οι τρεις πρώτοι λόγοι –που καλύπτουν το μεγαλύτερο ποσοστό (54,9%) των λόγων– αφορούν θέματα κακής οργάνωσης από την πλευρά των φοιτητών.

Σε διαφορετική έρευνα [8], πάλι παρουσιάζεται πως το κυριότερο πρόβλημα που αντιμετωπίζουν οι φοιτητές είναι η σωστή δόμηση του προγράμματός τους. Συνήθως ο τρόπος διαβάσματός τους καθοδηγείται από τις ίδιες τις εργασίες που έχουν να κάνουν, μιας και μόνο αυτές έχουν καταληκτικές ημερομηνίες παράδοσης, και έτσι παραμελούν τα υπόλοιπα καθήκοντα που έχουν, όπως το να παρακολουθούν τις διαλέξεις.

Όλα αυτά μας οδηγούν στο συμπέρασμα πως είναι απαραίτητος ένας αποτελεσματικός τρόπος προγραμματισμού και διαχείρισης των εργασιών τους.

2.2.2 Χαρακτηριστικά που οι φοιτητές θα επιθυμούσαν σε μια εφαρμογή

Σε έρευνα [8] που πραγματοποιήθηκε στο τμήμα Πληροφορικής του Πανεπιστημίου του Εδιμβούργου, διαπιστώθηκε πως η πλειοψηφία της ακαδημαϊκής κοινότητας επιθυμεί μια εφαρμογή διαχείρισης εργασιών να διαθέτει ημερολόγιο (θεώρησαν σημαντικό να είναι καταγεγραμμένες οι ημερομηνίες έναρξης/λήξης για κάθε εργασία για τη σωστή οργάνωση, όπως επίσης και χρωματική ταξινόμηση [color-coding] των εργασιών), ειδοποιήσεις / γνωστοποιήσεις για τις εργασίες και to-do λίστες (με προτεραιοποίηση, ομαδοποίηση και δυνατότητα εμφάνισης μπάρας προόδου). Επίσης εκφράστηκε ενδιαφέρον για τη δημιουργία ενός συστήματος ανταμοιβής, με σκοπό την ενθάρρυνση των φοιτητών να ολοκληρώνουν εργασίες.

Κεφάλαιο 3

Low code

Στις μέρες μας, όλα τα υπάρχοντα συστήματα σε κάθε τομέα, από τις τραπεζικές υπηρεσίες ως τα αεροναυπηγικά, ελέγχεται από κάποιο λογισμικό. Ως αποτέλεσμα τα λογισμικά γίνονται όλο και πιο περίπλοκα, με την ανάπτυξή τους να απαιτεί υψηλό κόστος και χρόνο.

Από το 2014 και μετά, αρχίζει και εδραιώνεται μια νέα προσέγγιση ανάπτυξης λογισμικού, που έχει ως άμεσο στόχο την αύξηση της παραγωγικότητας των προγραμματιστών. Οι πλατφόρμες αυτές ονομάζονται **Πλατφόρμες Ανάπτυξης Εφαρμογών σε Low-Code** (Low-Code Application Platforms - LCDP)¹. [1]

3.1 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, επιτρέποντας χρήστες χωρίς προγραμματιστική εμπειρία (citizen developers) 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] [7]

¹Εναλλακτικές ονομασίες είναι Low-Code Platforms (LCP), Low-Code Development Platforms (LCDP), ενώ η διαδικασία ανάπτυξης αναφέρεται ως Low-Code Software Development (LCSD).

“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.” [5]

3.2 Προηγούμενες προσπάθειες

3.2.1 Computer-Aided Software Engineering

Η πρώτη προσπάθεια αυτοματοποίησης της ανάπτυξης λογισμικού ήταν στη δεκαετία του 1990 με τη χρήση Computer-Aided Software Engineering (CASE) περιβαλλόντων τα οποία παρέχουν βοηθητικά εργαλεία και μεθόδους με σκοπό την πιο αποτελεσματική και χωρίς σφάλματα ανάπτυξη. Τα CASE περιβάλλοντα έθεσαν τα θεμέλια για τη δημιουργία νέων υποδειγμάτων, όπως ο οπτικός προγραμματισμός (visual programming) και ο προγραμματισμός που βασίζεται σε μοντέλα. [3]

3.2.2 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 παραπάνω στον σωστό διαχωρισμό των χαρακτηριστικών, στην αφαιρετικότητα και στην αυτοματοποίηση. [6] [2]

Κεφάλαιο 4

Mendix

Κεφάλαιο 5

<ανάπτυξη εφαρμογής>

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] Albert E Case. *Computer-aided software engineering (case): technology for improving software development productivity*. 1985.
- [4] 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.
- [5] Ibm. *What is low-code?* Αύγ. 2024. URL: <https://www.ibm.com/topics/low-code>.
- [6] Object Management Group - Model Driven Architecture. URL: https://www.omg.org/mda/faq_mda.htm.
- [7] 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.
- [8] Julia Castillo Trujillo. *Designing A Time Management App For And With Informatics Students*. 2020.