



## ΟΤΑΝ ΟΙ ΑΛΓΟΡΙΘΜΟΙ ΣΥΝΑΝΤΟΥΝ ΤΟΝ ΠΡΑΓΜΑΤΙΚΟ ΚΟΣΜΟ

Στα κείμενα της σειράς Contre Tech έχουμε ξανασχοληθεί με το θέμα της τεχνητής νοημοσύνης, αλλά αφού τα παρακολουθείτε ανελλιπώς τότε το ξέρετε. Να ενημερώσουμε λοιπόν εδώ ότι όποιος ψάχνει για κείμενα που θα λένε «όλη την αλήθεια» για το θέμα, μάλλον θα απογοητευτεί. Αντίθετα, αυτό που προσπαθούμε να κάνουμε είναι να κοιτάμε την τεχνητή νοημοσύνη από διάφορες πλευρές. Με αυτόν τον τρόπο κάθε φορά γινόμαστε και λίγο σοφότεροι.

Για τις ανάγκες του κειμένου αυτού υπενθυμίζουμε εν τάχει τι γράφαμε στο προηγούμενο τεύχος: λέγαμε ότι οι εφαρμογές του είδους ChatGPT βασίζονται στα λεγόμενα μοντέλα μηχανικής μάθησης. Ένα τέτοιο μοντέλο για να δημιουργηθεί πρέπει ένας αλγόριθμος να «εκπαιδευτεί» με πολλά, πάρα πολλά δεδομένα. Δυστυχώς όλη τη βιβλιογραφία του κόσμου που υπάρχει σε ψηφιακή μορφή και όλο το κείμενο που υπάρχει στο ίντερνετ. Ο αλγόριθμος δεν «μαθαίνει» λέξεις με την ανθρώπινη έννοια αλλά ανακαλύπτει στατιστικές συσχετίσεις μεταξύ λέξεων και φράσεων. Οι απαντήσεις που παράγει βασίζονται στην πρόβλεψη του ποιες λέξεις είναι πιο πιθανό να ταιριάζουν περισσότερο σε μια απάντηση που θα ικανοποιούσε τον χρήστη που δίνει το ερώτημα.

Γράψαμε επίσης και κάποιες διαπιστώσεις για τα μοντέλα. Ότι είναι στην πραγματικότητα έτσι όπως θα ήθελαν τα αφεντικά να είναι τα ανθρώπινα μυαλά μας. Λέγαμε συγκεκριμένα ότι τα μοντέλα «δεν μπορούν να διατυπώσουν τίποτα πρωτότυπο, είναι ανίκανα για οποιαδήποτε κριτική ερμηνεία των κειμένων και εξ' ορισμού εξαφανίζουν τις μειωψικές απόψεις. Στην ουσία αποτελούν μια τεχνική αποτύπωση της κυρίαρχης αλήθειας και μοντέρνα έκφραση του τεχνολογικού φετιχισμού». <sup>1</sup> Με άλλα λόγια, οι απαντήσεις που παράγουν καθορίζονται αυστηρά από την φύση των δεδομένων εκπαίδευσης.

Έχοντας αυτά υπ' όψιν, παρακάτω θα υποστηρίξουμε πως τα μοντέλα αυτά, αντίθετα με ότι συζητιέται, είναι φτιαγμένα για να τα πηγαίνουν καλά (και αυτό όχι πάντα) σε αποστειρωμένες συνθήκες εργαστηρίου. Αλλά ότι στο περίπλοκο περιβάλλον των πραγματικών ταξικών σχέσεων είναι πραγματικά για τα μπάζα.

### Το αποστειρωμένο περιβάλλον ως προϋπόθεση

Μοντέλα μηχανικής μάθησης όπως αυτό πίσω από το ChatGPT ονομάζονται γλωσσικά μοντέλα γιατί σύμφωνα με τους δημιουργούς τους είναι ικανά για κάποιου είδους χειρισμό της φυσικής γλώσσας. Στοιχειώδης κριτική ενασχόληση με τα εν λόγω εργαλεία κάνει σαφές το ποιες ακριβώς λειτουργίες μπορούν να φέρουν εις πέρας, ας πούμε πετυχημένα. Για παράδειγμα, μπορούν να σου αραδιάσουν εγκυκλοπαιδική πληροφορία που κανονικά θα έβρισκες στο wikipedia. Εδώ πέρα η πιθανοτική τοποθέτηση λέξεων μπορεί να παράξει μια απάντηση με κάποια χρησιμότητα γι' αυτόν που αναζητά πληροφορίες. Αντίστοιχα, με δεδομένο ότι τα μοντέλα έχουν εκπαιδευτεί με ακαδημαϊκό υλικό, μπορούν να λύσουν μια σχολική άσκηση και να περάσουν ένα πανεπιστημιακό τεστ. Τα καταφέρνουν δηλαδή σε ένα πλαίσιο που κάθε ερώτηση αντιστοιχεί σε απαντήσεις από μια καθορισμένη διδακτέα ύλη. Έπειτα, μπορούν ας πούμε να μεταφράσουν τις οδηγίες χρήσης ενός πλυντηρίου – μετάφραση που έτσι κι αλλιώς δεν προορίζεται να διεκδικήσει κανένα βραβείο ποιότητας. Κι εδώ θα αναρωτηθούμε. Η χρήση της φυσικής γλώσσας αφορά μόνο την στεγνή παράθεση πληροφορίας; Για δοκιμάστε να ζητήσετε στο ChatGPT να γράψει ή να μεταφράσει λογοτεχνικό κείμενο, να γράψει ποιήματα ή ρίμες, να παράξει πολιτικό κείμενο με επίκαιρη ταξική ανάλυση. Όλα αυτά σε κάποιου είδους πρωτότυπη μορφή και να μην είναι κονσέρβα. Ναι, το αποτέλεσμα θα είναι μεγάλη κριντζιά, αν το επιχειρήσετε μην πείτε ότι δεν σας προειδοποιήσαμε.

Οι δημιουργοί της τεχνητής νοημοσύνης παριστάνουν ότι μπορεί να δου-

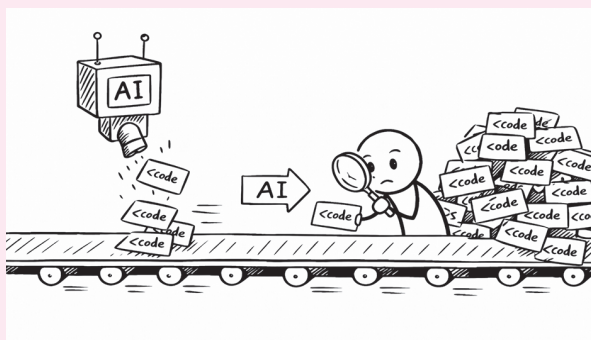
λέψει καλά υπό οποιοσδήποτε συνθήκες, ενώ στην πραγματικότητα λειτουργεί καλά σε αποστειρωμένο περιβάλλον δοκιμαστικού σωλήνα. Δεν είναι τυχαίο ότι ένα από τα αγαπημένα παραδείγματα όσων μιλάνε για τα κατορθώματά της αφορά το σκάκι. Το 2017 η εταιρεία DeepMind της Google, δημοσιοποίησε το AlphaZero, ένα πρόγραμμα τεχνητής νοημοσύνης το οποίο κατάφερε να κερδίσει σε παιχνίδια σκάκι τα καλύτερα σκακιστικά προγράμματα και να κάνει ακόμα και τον Κασπάροφ τον ίδιο να τρίβει τα μάτια του.<sup>2</sup> Ο αλγόριθμος του AlphaZero αναγνωρίζει μοτίβα κινήσεων σε ένα σύνολο πιθανοτήτων που το ανθρώπινο μυαλό δεν μπορεί να επεξεργαστεί και πολλές φορές θυσιάζει πιόνια που οι άνθρωποι παίκτες θα θεωρούσαν πολύτιμα. Η τεχνητή νοημοσύνη τα πηγαίνει καλά στο σκάκι ακριβώς επειδή το πλαίσιο του παιχνιδιού διέπεται από απαραβίαστους κανόνες και αυστηρά καθορισμένες κινήσεις των πιονιών. Η μη συμμόρφωση με αυτά καθιστά το παιχνίδι άκυρο. Ένας πύργος δεν επιτρέπεται να κινηθεί διαγώνια και αν φαγωθεί ο βασιλιάς τότε η παρτίδα τελειώνει. Σε ένα τέτοιο πλαίσιο το οποίο είναι απογυμνωμένο από απρόβλεπτα ενδεχόμενα, που το μόνο που διαφέρει κάθε φορά είναι η αλληλουχία κινήσεων των πιονιών, προφανώς το ανθρώπινο μυαλό υστερεί ενάντια σε έναν αλγόριθμο με υπολογιστική ισχύ ικανή για εκατομμύρια υπολογισμούς το δευτερόλεπτο.

### Οι μάντζερς φαντασιώνονται γραφεία χωρίς εργάτες

Ένα άλλο πεδίο στο οποίο η τεχνητή νοημοσύνη φαινομενικά τα πηγαίνει καλά είναι η γραφή κώδικα για υπολογιστές. Ο κώδικας αυτός καταχρηστικά αποκαλείται «γλώσσα» προγραμματισμού, στην πραγματικότητα όμως δεν έχει τις δυνατότητες της φυσικής γλώσσας. Μια γλώσσα προγραμματισμού στη σύγχρονη μορφή της αποτελείται από ένα αυστηρά καθορισμένο σύνολο «λέξεων» στα αγγλικά που συνοδεύονται επίσης από συντακτικούς κανόνες, και γενικά εκφράζει μια μαθηματική λογική. Στην ουσία είναι ένας τρόπος γραφής εντολών που είναι φτιαγμένες για να μεταφράζονται σε ενέργειες του υπολογιστή. Δεν πρόκειται δηλαδή για γλώσσα που επιτρέπει την απόκλιση από τους αυστηρούς κανόνες γραφής και αποκλείει την έκφραση πρωτότυπων ιδεών, καινούριων λέξεων ή τέλος πάντων οποιουδήποτε νοήματος εκτός του προκαθορισμένου πλαισίου χρήσης της. Με άλλα λόγια, οι γλώσσες προγραμματισμού αποτελούν ένα κλειστό σύστημα εντολών και συμβόλων που επιβραβεύει τον ψυχαναγκασμό και τιμωρεί τις ασάφειες για όποιον τις χρησιμοποιεί.

Κάπου εδώ λοιπόν αναδύεται, όπως και στην περίπτωση της παρτίδας σκάκι, ένα πλαίσιο όπου το μοντέλο μηχανικής μάθησης φαίνεται να υπερτερεί του ανθρώπινου εγκεφάλου. Το μοντέλο έχει εκπαιδευτεί με όλες τις γλώσσες προγραμματισμού, με τους κανόνες τους και τη μαθηματική λογική τους, και έχει τη δυνατότητα να παράξει τον κώδικα που θα του ζητηθεί μέσα σε λίγα δευτερόλεπτα. Επιπλέον, πέρα από τη γλώσσα προγραμματισμού καθαυτή, έχει εκπαιδευτεί με ολόκληρη την ποσότητα τεχνικής πληροφορίας η οποία απαιτείται για τη δημιουργία εφαρμογών, από την οποία ένα μικρό κλάσμα μόνο μπορεί να συγκρατήσει η ανθρώπινη μνήμη: εδώ πέρα εννοούμε εφαρμογές, κώδικα και εργαλεία που δημιουργούνται με μοναδικό σκοπό να αποτελέσουν τα εργαλεία δουλειάς ενός προγραμματιστή. Διότι, ένα σημαντικό κομμάτι της δουλειάς του προγραμματιστή είναι η διαδικτυακή αναζήτηση πληροφορίας, τεχνικών προδιαγραφών και εγχειριδίων για να βρει πως δουλεύει κάτι ή να λύσει ένα πρόβλημα. Ε λοιπόν, αφού το AI δημιουργεί κώδικα σε δευτερόλεπτα, έχοντας στη διάθεσή του όλη την τεχνική πληροφορία του κόσμου και τους κανόνες γραφής μιας γλώσσας που δεν μπορείς να χρησιμοποιήσεις για να πεις τη μαλακία σου, τότε τι μας εμποδίζει να το ανακηρύξουμε προγραμματι-

«ΜΕ ΒΟΗΘΑΕΙ ΑΛΛΑ ΔΕΝ ΜΠΟΡΩ ΝΑ ΚΑΤΑΛΑΒΩ  
ΠΩΣ ΘΑ ΤΟ ΚΑΝΩ ΝΑ ΜΕ ΒΟΗΘΗΣΕΙ ΠΡΑΓΜΑΤΙΚΑ ΠΟΛΥ»



Το παραπάνω είναι η ψιλοαπελπισμένη διαπίστωση ενός προγραμματιστή που παλεύει να είναι πιο παραγωγικός χρησιμοποιώντας το AI.<sup>7</sup> Η τεχνητή νοημοσύνη έσκασε στους χώρους εργασίας με τυμπανοκρουσίες και τιρινίνι. Οι μάντζερς στα σοβαρά πιέζουν να χρησιμοποιείται παραπάνω, κατηγορώντας πολλές φορές τους υπαλ-

λήλους ότι δεν τη χρησιμοποιούν όσο πρέπει.<sup>8</sup> Οι προγραμματιστές υποφέρουν από FOMO για να μην μείνουν πίσω στις τεχνολογικές εξελίξεις και πιέζονται να βρουν τρόπους χρήσης του AI στην καθημερινή τους δουλειά. Το αμερικανικό κράτος που έχει επιβάλει το AI και το FOMO σε όλους τους υπόλοιπους, τρίβει τα χέρια του. Όλοι μαζί παριστάνουν ότι η παραγωγικότητα της εργασίας είναι ατομικό ζήτημα του κάθε μεμονωμένου υπαλλήλου. Τα αποτελέσματα πολλές φορές είναι κωμικοτραγικά. Σύμφωνα με έναν προγραμματιστή, πριν το AI η δουλειά του ήταν να γράψει τον κώδικα, να τον τεστάρει και να τον ανεβάσει. Τώρα έχει γίνει μια μπερδεμένη αλληλουχία βημάτων: φτιάξε την εντολή για το AI, περίμενε, διάβασε τον κώδικα που δημιούργει, αξιολόγησε αν ο κώδικας είναι σωστός, ασφαλής και ταιριάζει με τον υπόλοιπο κώδικα της εφαρμογής, ξαναφτιάξε εντολή για το AI για να κάνει διορθώσεις και φτου κι απ' την αρχή. Όλα αυτά με μια αίσθηση δυσπιστίας ότι ο κώδικας που φτιάχνει το AI δεν είναι επιπλέον άρα θέλει πολύ πιο προσεκτικό έλεγχο.<sup>9</sup> Οπότε το εργαλείο που θα σου εξοικονομούσε χρόνο σου τρώει πολλές φορές περισσότερο χρόνο για να εξοικειωθείς με τα καπρίτσια του.

στή της χρονιάς και να του δώσουμε και όσκαρ; Τι να τους κάνουμε τους προγραμματιστές ενώ στην πραγματικότητα χρειαζόμαστε χειριστές μοντέλων μηχανικής μάθησης, να τα ταΐζουν εντολές σε φυσική γλώσσα και αυτά να φτύνουν κώδικα;

Το ζήτημα του αυτοματισμού στο γραφείο, όπως σίγουρα έχετε παρατηρήσει, αποτελεί ένα από τα πιο καυτά θέματα στον δημόσιο λόγο. Η εξαφάνιση της υπαλληλικής εργασίας προαναγγέλλεται με ψυχραιμες διατυπώσεις όπως αυτή του διευθυντή του τμήματος τεχνητής νοημοσύνης της Microsoft. Σύμφωνα με τα λεγόμενά του όλες οι δουλειές γραφείου θα αντικατασταθούν από την τεχνητή νοημοσύνη μέσα σε 18 μήνες.<sup>3</sup> Όμως, αυτού του είδους η αφεντικότητα ρητορική, όσο κι αν παριστάνει ότι αποτελεί κάτι μοντέρνο, δεν είναι πρωτοφανής στην ιστορία του καπιταλισμού.

Ας πάμε, για παράδειγμα, κάμποσες δεκαετίες πίσω. Το 1959 δημιουργήθηκε η γλώσσα προγραμματισμού COBOL, σαν ένα σύνολο εντολών υπολογιστή σε φυσική γλώσσα. Οι δημιουργοί της προέβλεπαν ένα μέλλον όπου η χρήση υπολογιστή στο εταιρικό περιβάλλον και οι δυνατότητες που έδινε η COBOL, θα ενίσχυαν την παραγωγικότητα αντικαθιστώντας την επεξεργασία δεδομένων στο χαρτί. Μαζί με αυτό, ο έμφυλος προσδιορισμένος ρόλος της γραμματέως θα μπορούσε μέχρι και να εκλείψει, αφού ο υπολογιστής θα εκτελούσε γρηγορότερα τις εργασίες που μια γραμματέας έφερνε εις πέρας με το χέρι. Είκοσι χρόνια μετά οι γραμματείς δεν είχαν εξαφανιστεί. Αντίθετα, ο ρόλος τους είχε μετασχηματιστεί ώστε να πειθαρχήσουν σε ένα νέο καταμερισμό της εργασίας που είχε φτιαχτεί με κέντρο τον υπολογιστή. Οι γραμματείς καλούνταν πλέον να λειτουργήσουν σαν σύνδεσμοι ανάμεσα στους μάντζερς και στο κέντρο υπολογιστών, μεταφέροντας πληροφορίες με τρόπο που να υπακούει στις προδιαγραφές που αυστηρά ορίζονταν από τους προγραμματιστές. Ο ρόλος τους παρέμεινε νευραλγικός στην ομαλή λειτουργία του γραφείου.<sup>4</sup>

### Τα περίπλοκα συστήματα βάζουν δύσκολα στους αλγόριθμους

Οι προγραμματιστές της συντακτικής ομάδας του περιοδικού μπορούμε να σας διαβεβαιώσουμε ότι πράγματι τα μοντέλα, όσον αφορά το επίπεδο της γραφής του κώδικα, έχουν γίνει ένα διαδεδομένο εργαλείο στα τμήματα πληροφορικής των εταιρειών. Οι μάντζερς ενθαρρύνουν εντόνως

να χρησιμοποιούνται ώστε οι προγραμματιστές να βρίσκουν τρόπους να είναι πιο παραγωγικοί. Όλο αυτό φυσικά έχει κόστος – πολλές εταιρείες πληρώνουν μεγάλες συνδρομές στις αμερικανικές εταιρείες που φτιάχνουν τα μοντέλα για να έχουν πρόσβαση στις πιο εξελιγμένες εκδόσεις τους. Για τους λόγους που αναφέραμε παραπάνω, η χρήση των μοντέλων βοηθάει σε διάφορες όψεις της καθημερινής δουλειάς μιας ειδικότητας που η λογική της αυτοματοποίησης ανήκει στις βασικές αρχές της. Ας πούμε, χρησιμοποιούνται σαν ένα αναβαθμισμένο google search: αντί για την αναζήτηση μιας λύσης σε ένα πρόβλημα που έχει ποστάρει κάποιος άλλος προγραμματιστής σε κάποιο φόρουμ, το μοντέλο σου δημιουργεί μια πιθανή λύση γρηγορότερα. Μπορούν να κάνουν μια χαμηλής περιπλοκότητας αλλά χρονοβόρα επεξεργασία κώδικα, να εξετάσουν των κώδικα για πιθανές δυσλειτουργίες ή μπορούν να βοηθήσουν τον προγραμματιστή να καταλάβει τη λειτουργία του κώδικα που έχει γράψει κάποιος άλλος. Φυσικά, όλα αυτά δεν τα κάνουν πάντα σωστά.

Αλλά τα πράγματα δεν είναι τόσο απλά και ιδανικά, όπως τίποτα άλλωστε σ' αυτή τη σκατοζωή. Διότι ο κόσμος της παραγωγής λογισμικού δεν είναι σαν το αποστειρωμένο περιβάλλον γραφής κώδικα του κάθε μεμονωμένου προγραμματιστή.

Κατ' αρχάς, ο κώδικας που θα γραφτεί από ένα μοντέλο μπορεί να είναι συντακτικά σωστός και αυτό όχι πάντα. Αλλά τίποτα δεν εξασφαλίζει ότι ο κώδικας θα είναι λειτουργικός κατά την εκτέλεση της εφαρμογής. Πολύ συχνά παράγεται κώδικας που δεν δουλεύει όπως θα ήθελε ο προγραμματιστής ή δημιουργεί σφάλματα στην λειτουργία της εφαρμογής. Όσο πιο περίπλοκο το αίτημα προς το μοντέλο, τόσο πιο πιθανό η λύση που θα προτείνει να μην είναι λειτουργική. Μάλιστα διάφοροι προγραμματιστές τονίζουν ότι η φαινομενική ευκολία που προσφέρουν οι εφαρμογές αυτές στον γρηγορότερο προγραμματισμό, μπορεί να δημιουργήσουν δυσκολίες κατά τη συντήρηση μιας εφαρμογής στο μέλλον λόγω της περιπλοκότητας ή της χαμηλής ποιότητας του κώδικα που παράγουν.<sup>5</sup> Οι σύγχρονες εφαρμογές αποτελούνται από εκατομμύρια γραμμές κώδικα, γραμμένες από εκατοντάδες διαφορετικούς ανθρώπους και δεκάδες διαφορετικά υποσυστήματα που πρέπει να δουλεύουν στοιχειωδώς καλά ώστε να μπορεί η εφαρμογή να εκπληρώσει τον δεδηλωμένο σκοπό της. Πρόκειται για συ-

## Ο ΧΑΟΤΙΚΟΣ ΚΟΣΜΟΣ ΤΗΣ ΠΑΡΑΓΩΓΗΣ ΛΟΓΙΣΜΙΚΟΥ



Ακολουθεί παράδειγμα βγαλμένο από έναν αληθινό χώρο εργασίας από αυτούς που γίνεται η παραγωγή λογισμικού. Ο νέος κώδικας που γράφει ένας προγραμματιστής, πριν γίνει διαθέσιμος μέσω update στην εφαρμογή που χρησιμοποιούν οι πελάτες, πρέπει πρώτα να τεσταριστεί ότι είναι λειτουργικός. Αυτό συμβαίνει σε μια δοκιμαστική έκδοση της εφαρμογής που δημιουργείται αποκλειστικά για εσωτερική χρήση. Σε αυτήν, ένας άλλος υπάλληλος που συνήθως αποκαλείται tester, θα ελέγξει ότι η λειτουργικότητα είναι αυτή που πρέπει και θα βρει ενδεχόμενα σφάλματα που πρέπει να διορθωθούν.

Τα δοκιμαστικά περιβάλλοντα της εταιρείας όμως χρησιμοποιούνται από δεκάδες διαφορετικούς προγραμματιστές μέσα στη μέρα. Πολλοί διαφορετικοί άνθρωποι ρίχνουν εκεί τον κώδικά τους για να τεσταριστεί. Το αποτέλεσμα είναι ότι τα δοκιμαστικά περιβάλλοντα συχνά είναι μη λειτουργικά. Στην περίπτωση μας, ένας φουκαράς tester προσπαθεί για 4 μέρες να ρυθμίσει ένα τέτοιο περιβάλλον για να κάνει τη δουλειά του αλλά το περιβάλλον δεν συνεργάζεται. Ο

tester αναζητά κάποιον συνάδελφο να τον βοηθήσει και να του δώσει πληροφορία που ενδεχομένως του διαφεύγει για να ρυθμίσει σωστά το περιβάλλον. Μιλάει με αρκετούς διαφορετικούς ανθρώπους. Βρίσκει μια άκρη και ξεκινάει να τεστάρει τον κώδικα του προγραμματιστή. Ανακαλύπτει ένα λεγόμενο bug – σφάλμα στη λειτουργία της εφαρμογής. Το συζητάει με τον προγραμματιστή. Ο προγραμματιστής διαπιστώνει ότι το bug οφείλεται σε κώδικα που έχουν γράψει προγραμματιστές από άλλο τμήμα της εταιρείας. Ακολουθούν συζητήσεις με τους προγραμματιστές από το άλλο τμήμα. Αυτοί συμφωνούν ότι είναι δικό τους το λάθος και με βάση το πρόγραμμά τους, θα έχουν έτοιμο τον διορθωμένο κώδικα σε δύο εβδομάδες.

Τελικά για παρόμοιους λόγους θα χρειαστούν άλλες δύο βδομάδες μετά τις δυο βδομάδες για να μπορέσει να πραγματοποιηθεί το τεστάρισμα της εφαρμογής. Ίσως το τεστάρισμα ολοκληρωθεί επιτυχώς χωρίς να βρεθούν bugs, ίσως και όχι, οπότε προσθέστε καμιά βδομαδούλα ακόμα. Η παραγωγή λογισμικού στο πλαίσιο μια μεγάλης εταιρείας απαιτεί την συνεργασία εκατοντάδων υπαλλήλων, χωρίς να υπολογίζονται όσοι δουλεύουν για να είναι λειτουργικά τα εξωτερικά συστήματα που χρησιμοποιούνται ως σταθερό κεφάλαιο. Αυτό απαιτεί χιλιάδες συνεννοήσεις μέσα στην εργάσιμη μέρα, μεταφορά πληροφορίας από το ένα τμήμα στο άλλο, συγκεκριμένη οργάνωση της εργασίας και επιτήρηση της λειτουργίας της με τεχνικούς τρόπους. Δηλαδή ένα πλαίσιο που μπορεί να προκύψουν ένα κάρο απρόβλεπτα ενδεχόμενα: ασυνεννοησία, συστήματα που υπολειπονται, bugs, προγραμματιστές που λουφάρουν, κυβερνοεπιθέσεις, πλοία του ρωσικού σκιάδους στόλου που κόβουν τα υποθαλάσσια καλώδια ενός σέρβερ στην άλλη άκρη του πλανήτη και άλλα τέτοια. Μας λένε στα σοβαρά ότι το AI είναι ικανό να τα βγάλει πέρα στον χαοτικό κόσμο της εκμετάλλευσης της εργασίας. Κι εμείς τους πιστεύουμε σαν μαλάκες.

στήματα τόσο ευαίσθητα σε σφάλματα που μια μικρή αλλαγή σε μια γραμμή κώδικα μπορεί να προκαλέσει την κατάρρευσή τους.

Έπειτα, η άποψη ότι το AI θα αυξήσει την παραγωγικότητα της εργασίας των προγραμματιστών συσκοτίζει το γεγονός ότι η ταχύτητα που παράγεται το λογισμικό δεν έχει να κάνει με το πόσο γρήγορα γράφεται ο κώδικας αλλά κυρίως με το πόσο καλά δουλεύει ο καταμερισμός της εργασίας που έχει δομηθεί γύρω από τον προγραμματιστή. Τα τμήματα τεχνολογίας των εταιρειών στελεχώνονται κατ' αρχάς από προγραμματιστές με αρκετές διακριτές εξειδικεύσεις και χωρισμένους σε ομάδες, που η καθεμία εργάζεται συνήθως πάνω σε ένα συγκεκριμένο υποσύστημα μιας εφαρμογής. Ένα μεγάλο σύνολο άλλων υπαλλήλων δουλεύει μαζί με τους προγραμματιστές κάνοντας διαφόρων ειδών συμπληρωματικές εργασίες: συντάσσουν τις προδιαγραφές της εφαρμογής, φροντίζουν για την επικοινωνία μεταξύ των διαφορετικών ομάδων, επιβλέπουν την πρόοδο και τα χρονοδιαγράμματα των πράξεων, συντηρούν τα συστήματα, τεστάρουν ότι ο κώδικας δουλεύει όπως πρέπει, επικοινωνούν τα τεχνικά προβλήματα των πελατών και ένα σωρό άλλα. Η ομαλή λειτουργία όλου αυτού του καταμερισμού κάθε άλλο παρά δεδομένη είναι και εξασφαλίζεται με μεθόδους μάντζμεντ και τη χρήση κάμποσων εφαρμογών που χρησιμοποιούνται σαν εργαλεία αυτοματισμού διαδικασιών και επιτήρησης. Αν συνυπολογίσουμε μάλιστα, ότι οι εταιρείες τεχνολογίας (στην προκειμένη οι ελληνικές) βασίζονται σε εξωτερικά συστήματα, εφαρμογές και τεχνολογίες αμερικάνικων εταιρειών,<sup>6</sup> τότε αυτός ο καταμερισμός της εργασίας στην πραγματικότητα αφορά την ομαλή λειτουργία ολόκληρου του διεθνούς εργοστασίου, που με τη σειρά του βασίζεται πάνω στο σύστημα

διεθνών διακρατικών σχέσεων, που... τρέχα γύρευε πώς θα λειτουργεί τα επόμενα χρόνια.

Για το κλείσιμο, κρατήστε το εξής. Όταν ακούτε ότι η εφαρμογή που φτύνει κώδικα, από μόνη της θα αυξήσει την παραγωγικότητα της εργασίας, κρατήστε μικρό καλάθι. Διότι η παραγωγικότητα, όσο απασχολεί διακαώς τα αφεντικά και τους μάντζερς, άλλο τόσο δεν είναι σκέτα τεχνικό ζήτημα, αλλά πάνω απ' όλα πολιτικό. Αν συμβεί τέτοια αύξηση της παραγωγικότητας στην εποχή μας, θα συμβεί επειδή ένας νέος καταμερισμός της εργασίας θα έχει φτιαχτεί. Τόσο περίπλοκος όσο και παγκόσμιος. Που θα περιλαμβάνει εργασία που προσεγγίζει τη σκλαβιά και εξειδικευμένη εργασία με πολύπλοκα εργαλεία αυτοματισμού – ανάλογα σε ποιο σημείο του πλανήτη θα στρέψει κανείς το βλέμμα του. Πάνω απ' όλα θα έχει να κάνει με το πώς ο εργάτης θα πειθαρχήσει στον καταμερισμό της εργασίας που θα του επιβληθεί. Διαδικασία που περιλαμβάνει βία, καταναγκασμό και άλλες τέτοιες πανάρχαιες μεθόδους μάντζμεντ.

1. «Γυρεύουν τα κράτη να φτιάξουν έξυπνες μηχανές;», *Antifa* #98, 01/2026
2. Henry Kissinger, Eric Schmidt, Daniel Huttenlocher, *Η εποχή της τεχνητής νοημοσύνης και το ανθρώπινο μέλλον μας*, Liberal Books, 2022
3. «Όλες οι δουλειές γραφείου θα αντικατασταθούν από την τεχνητή νοημοσύνη μέσα σε 18 μήνες», *kathimerini.gr*, 02/2026
4. David E. Dunning, *Autocoding at work: designing language, specifying labor*, University of Pennsylvania, 01/2026
5. «Ai coding is now everywhere. But not everyone is convinced.», *technologyreview.com*, 12/2025
6. Γυρεύουν τα κράτη να φτιάξουν έξυπνες μηχανές;», ο.π.
7. «Ai coding is now everywhere. But not everyone is convinced.», ο.π.
8. «Claude code and the great productivity panic of 2026», *bloomberg.com*, 2/2026
9. «Ai fatigue is real and nobody talks about it», *siddhantkhar.com*, 2/2026