

Katowice, dn. 1 marca 2024 r.

**Recenzja rozprawy doktorskiej Pani Magister Anny Paprzyckiej
pt. „Oprogramowanie jako system operacyjny kultury. Między historią a praktykami
współczesności”**

Gdyby myśleć o wzorowym przedstawieniu oprogramowania jako przedmiotu namysłu krytycznego – wraz z objaśnieniem jego kulturowej historii i kontekstów – niniejsza praca z pewnością posłużyłaby za model. Nie tylko bowiem dobrze objaśnia znaczenia oraz pojęcia oprogramowania na przestrzeni jego rozwoju, lecz również sprawnie mapuje teoretyczne ujęcia *software studies*, organizując je w logicznych porządkach tak diachronicznych jak i synchronicznych względem rozwoju zaawansowanych technologii.

Praca liczy sobie 246 stron, wraz ze streszczeniami i bibliografią obejmującą 254 trafnie dobrane źródła. Świadczy to o dobrym osadzeniu podjętej dyskusji nad oprogramowaniem jako systemem operacyjnym kultury w kontekstach krajowych i zagranicznych, jak również o spełnieniu oczekiwań stawianych pracom doktorskim; należy do nich między innymi umiejętność dialogu z literaturą przedmiotu oraz uplasowanie podjętego namysłu względem istniejących dyskusji krytycznych, co praca spełnia w sposób zadawalający.

Mogę więc już teraz pogratulować Autorce dobrej dysertacji ale przede wszystkim poprowadzenia wywodu po „szerszych orbitach”. Praca nie ogranicza się bowiem do zjawisk samego oprogramowania, ale wychodzi w kierunku hiperobiektyw, jak *media* czy *technologia*, względem których definiuje software podług wybranych perspektyw.

Obierając za przedmiot swojej analizy organizujący potencjał oprogramowania – obecny tak w środowiskach technologicznych jak i w zjawiskach społecznych, politycznych i ekonomicznych – praca dywaguje nad technikami treści oraz przekazu (*technics of content; technics of communication*). Pyta o naturę i uzus urządzeń interaktywnych, porównuje je z naturą i uzusem mediów komunikacyjnych, wreszcie draży ich kondycję oraz rolę w świecie sprzed oprogramowania, to jest, sprzed epoki mediów obopólnych (*reciprocal media*), mediów natychmiastowych (*immediate media*), rządzących się innym pojęciem czasu i przestrzeni, oraz innym doświadczeniem materialności. Mowa tu dokładnie o mediach, w których procesy kulturowe zachodzą tak odgórnie





(*top-down*; tj. od strony projektowania) jak i oddolnie (*bottom-up*, tj. z kierunku użytkowania), jak również dwukierunkowo: od kodu do feedbacku i z powrotem, w warunkach przyspieszonej równoczesności (*accelerated media*).

Podjmując wszystkie te kierunki, praca od początku dobrze opowiada oprogramowanie, zwłaszcza w jego załączkach i skutkach. Widać to przede wszystkim w sposobie, jaki praca łączy procesy kultury cyfrowej z procesami technologii przed-cyfrowych oraz jak unaocznia aspekty technologiczne oprogramowania, pochodzące sprzed „ery” komputerowej. Ponadto, dobrze wyjaśnia przyczynki i inspiracje w obrębie software’u, ukazując oprogramowanie jako wynik skłonności organizacyjnych kultury a nie (lub nie tylko) „skłonności” w obrębie rozwoju technologii jako takiej. Dzięki temu, skonstruowana przez panią Annę Paprzycką wykładnia roli oprogramowania słusznie komplikuje namysł przyczynowo skutkowy nad rzeczywistością technologiczną. W sposób oczywisty, komplikuje tym samym namysł nad kulturową emanacją elementów software’u – w czym znaczącą rolę pełni stawiany przez Autorkę nacisk na ekspresywny charakter oprogramowania i rozszerzenie jego funkcji o semiotyczne konteksty. Software to – jak artykułuje Autorka – zjawisko czy zespół zjawisk, osadzanych w semiotycznie płynnym krajobrazie, który udając stabilność, ulega ciągłym przeobrażeniom, także pod wpływem „procesów programowych” (czy programistycznych: *software processes*) – rozumianych informatycznie i kulturowo.

Tym bardziej więc pochwalam przyjęte w pracy pojęcie oprogramowania jako „wielopoziomowej, hierarchicznej struktury”. I choć Autorka przyjmuje je w sposób typowy dla *software studies*, sposób w jaki to robi otwiera oprogramowanie na głębszą analizę, zwłaszcza użytkową. Świadczy o tym chociażby moment w rozdziale drugim, kiedy Autorka daje się poznać jako pragmatyczka i eksperymentatorka oprogramowania. Jest to moment kiedy odsłania swoje realizowane artystycznie doświadczenia z programami Quine, nastawione na produkcję kodów niefunkcjonalnych i samo referencyjnych: a więc nieużytecznych w antropomorficznym zamyśle „zadania i celu”.

Eksperymenty artystyczno-naukowe Pani Anny Paprzyckiej nawiązują do oprogramowania jako języka zaprzęgniętego w określoną funkcję społeczną i uzależnionego od ludzkich kontekstów. Tym samym starają się, niemal po hackersku, wynieść oprogramowanie poza ich użytkowy cel. Niestety czynią to wyłącznie w ramach określonego dekorum. Tym, co bowiem zaskakuje w przedstawionej przez Autorkę sztuce własnej jest dążność do utrzymania, by nie powiedzieć wzmocnienia, sprawczości człowieka względem organizacyjnego potencjału systemów. Jak gdyby





utrata funkcji w oprogramowaniu, a co za tym idzie, jego potencjalna autonomia były nie do przyjęcia. Ta sama kategoryczność przejawia się w podjętej przez Panią Annę próbie przewartościowania rekurencji software'u, którą to próbę ocenię jako nie tyle chybiona co nieco niepełną. Pozwala mi na to przede wszystkim brak uwzględnienia w niej rekursywności jako właściwości obiektów technicznych (w tym oprogramowania). Właściwość ta jest o tyle kluczowa, że uznaje w rekurencji czynnik „przygodny”, widząc „przygodność” w kategoriach przymiotu środowisk tak organicznych jak nieorganicznych, tak ludzkich jak i technologicznych. Uzdrawiające w tym miejscu byłoby zatem odwołanie do prac chińskiego filozofa Yuka Hui na temat kosmotechnicznego pluralizmu. Rażąco zabrakło ich w przedstawionej rozprawie, a ich obecność mogłaby bez wątpienia stonować nieco kategoryczny pogląd Autorki, że „nie istnieje coś takiego, jak oprogramowanie odizolowane od kontekstu społecznego” (s. 103), które pozwolę sobie tymczasem ostudzić kategorycznym: „Yet!” (mówiąc z angielska).

Pragnę jednak w tym miejscu, raz jeszcze podkreślić wartość zaangażowanego wymiaru pracy. Dzięki niemu, Autorka, jako badaczka nurtu *art-based-research*, w ramach własnej praktyki artystycznej jako relacji badawczej – nierzadko pomijanej w tradycji *software studies* – eksploruje oprogramowanie *pozasystemowo*, zgłębia kulturowy potencjał technologii informacyjnych i przekracza mocno ugruntowany aksjomat, wyrażany w przekonaniu, że badanie oprogramowania wymaga specjalistycznej wiedzy informatycznej. Uwiarygadnia tym samym pogląd nowojorskiej szkoły medioznawstwa, że oprogramowanie jest efektem i pośredniczeniem (tak wewnątrz procesów technologicznych jak i względem nich). Wzmocnieniem tego poglądu byłoby z pewnością odwołanie Henrego Jenkinsa i jego pojęcia kultury partycypacji (*participatory culture*), którego w pracy niestety brak. Pojęcie to, choć nienowe, spajałoby kulturowy mechanizm software'u, osadzając go na tle praktyk sieciowych, *feedbackowych*, oraz uzusu spod znaku „do it yourself together”, który, jak podkreśla Autorka, ma obecnie kluczowe znaczenie dla rozwoju oprogramowania. Odwołanie do kultury partycypacji pozwoliłoby także lepiej oddać kulturotwórczą moc procesów programowych. Na pewno umożliwiłoby robienie tego bez konieczności rozgraniczania na „prymitywny zachwyty nad kodem jako bytem magicznym i źródłem przyczynowej cyfrowości a oddziaływaniem społecznym kodu jako jego właściwej mocy” (tu odwołuję się do słów Autorki ze s. 25).

Mam spory problem z takim rozgraniczeniem. Jest w nim pewna stronniczość i swego rodzaju teoretyczna nieściśłość. Autorka przywołuje je w pracy za Wendy H. K. Chun, jednak bez krytycznego komentarza. Choć myślę, że warto, by w którymś miejscu, spróbowała je sprostować





lub lepiej omówić. Szczególnie, że chce szeroko mówić o sile uprzedzeń jako uwarunkowaniu ludzkich relacji z technologiami. Ten aspekt, to swoją drogą jeden z najciekawszych wątków wywodu, dobrze nakreślony ale wymagający lepszego rozwinięcia. Jak bowiem wynika z nie tak dawnej wienerowskiej wykładni, w popularnej i technokratycznej wyobraźni, technologie informatyczne uchodziły za prawie magiczne, bo przybierały charakter niemal odkupieńczy względem dotychczasowej materialnej dekadencji. Podsyczał to oczywiście heroizm „mit-informacji,” w którym systemy komunikacyjne, dane, oprogramowanie oraz złożone organizacje techniczne stawały się awatarami dobra, zastępującymi ulotny przekaz czymś znaczeniowoczym a zatem znaczącym. Sam Wiener kojarzył zachowania systemów informatycznych z wartością i życiem, nadając im walory niemalże boskie. Mistykę tę napędzał dodatkowo dyskurs biologiczny lat 50tych, zachłyśnięty odkryciami DNA i pojęciem kodów genetycznych, które, tłumaczyły je jako podstawę życia oraz ludzkiej łączności z kosmosem, czyniąc kod systemem nadawczo-odbiorczym czy nawet kosmicznym alfabetem. W tym rozumieniu, kod stawał się medium, a dekodujący i rekodujący go odbiorcy uczestnikami misterii o statusie magii. I znów, by nie pogłębiać skrótów myślowych, pozwolę sobie odesłać do książki *Profiles of the Future*, Arthura C. Clarke’a a także do *Techgnosis* Ericka Davisa, których nieobecność w tej pracy nieco zaskakuje i wymaga zadośćuczynienia – chociażby ze względu na przeglądowy charakter zaproponowanego wywodu.

Zaskakuje też brak dwóch istotnych wątków dla zachowania ciągłości namysłu nad formami oprogramowania i rolą oprogramowania w szerokich kontekstach społecznych. Mam tu na myśli software po automatyzacji oraz pedagogikę oprogramowania. Jeśli chodzi o pierwszy, praca co prawda opisuje etapy technologicznego zaawansowania, z których da się wyróżnić etap automatyzacji (czyli etap urządzeń i mechanizmów) a także etap post-automatyzacji (czyli etap sieci oraz AI), ale nie rozróżnia ich problemów. Co za tym idzie, nie wchodzi w dialog z tymi ujęciami, które polemizują z pozytywistycznymi odczytaniem Heideggera, skłaniającymi do porzucenia technologii na rzecz naprawy świata a nie ich odpoznania. Jeśli chodzi o wątek drugi, praca co prawda porusza alternatywne zastosowania w zakresie organizacji oprogramowania w kontekstach społecznych lecz robi to retrospektywnie a nie spekulatywnie. Innymi słowy nie zagłębia się w edukacyjny (by nie powiedzieć *farmakoniczny*) potencjał software’u oraz wpisany w jego „strukturę” element emancypacyjny. W obu przypadkach, z pomocą przyszedłby Bernard Stiegler – teoretyk i empiryk kondycji technologicznej, którego wsparcie nad tworzeniem emancypacyjnych nowych oprogramowań oraz nad emancypacją oprogramowań istniejących, mogłoby posłużyć Autorce za





dobrą wykładnię. Zwłaszcza, że znacząca część jego współpracy z IRI (*Institut de recherche et d'innovation du Centre Pompidou*) to właśnie motywacja nowego rodzaju software'u w myśl założeń o oprogramowaniu jako systemie operacyjnym kultury. Zaznaczał w niej, że zamiast porzucać struktury technologiczne, powinniśmy raczej przeformułowywać ich cel i formułować go na nowo. Tylko w ten sposób możliwe jest tworzenie zdrowych społeczeństw w dobie cyfrowej schizofrenii, która wymaga przemyśleń nie tylko z poziomu projektowania ale także z poziomu metafor oprogramowania oraz samego języka.

Problem ten wzywa do przemyślenia szerszej gamy software'owych zagadnień, takich jak przystępność oprogramowania (*software approachability*), symulacyjność oprogramowania (*software design foresight*), wreszcie semiotyka oprogramowania (*software semiotics*), których mi tutaj zabrakło, lub które utonęły w morzu wykładni medioznawczej albo nieco się w niej rozmyły. Podobnie zresztą jak rozmyła się historia użytkowania i ideologizacji oprogramowania, omówiona w pracy teorią szkoły frankfurckiej, z pominięciem J. M. Balkina i jego przewrotnej tezy o ideologii jako oprogramowaniu kultury. Intuicja podpowiada mi, że teza ta mogłaby pomóc Autorce mówić jeszcze szerzej i sprawniej niż ma to miejsce w przedstawionym wywodzie. To samo dotyczy „prastarej” teorii mediów masowych Dwighta MacDonalda, która lepiej niż teoria krytyczna sprzed eksplozji globalnych procesów medialnych, pozwoliłaby Autorce usytuować software względem przemian w zakresie kreacji, twórczości, autorstwa i innych zjawisk, jakie przyniósł rozwój mediów w dwudziestym wieku.

Powyższe uwagi to nie tyle zarzuty co wskazówki. Zwłaszcza jeśli niniejsza praca miałaby przerodzić się w książkę do czego zachęcam, ale wyłącznie po wprowadzeniu zmian. Te zmiany musiałyby dotyczyć tak luk i nieścisłości, które należałoby uzupełnić, jak znamion stylistycznych właściwych dysertacji, które z kolei należałoby usunąć. Zalicza się do nich przede wszystkim metatekst, którego swoją drogą jest gdzieniegdzie za dużo, a także powtórzenia (szczególnie s. 21) i tautologie (np. zdanie „Oprogramowanie nie jest prostym do zdefiniowania przedmiotem badań”, s. 9), będące spuścizną postmodernistycznej manieri.

Uzupełnienia wymagałyby przede wszystkim fragment o automatach, w którym zalecam sięgnąć do technologii hydraulicznych świata arabskiego sprzed naszej ery. Wymagałaby go także definicja oprogramowania w ujęciu Brattona. Oprogramowanie jako *stack* to nie tylko struktura na poziomie geologicznym; to także odzwierciedlenie właściwości świata fizycznego, a zatem tak wyrażenie współczesnej kondycji antropocenu jak i jej przekroczenia. Innymi słowy, niczym



bumerang powraca (pominięta w pracy) perspektywa pedagogiczna i emancypacyjna właściwość technologicznej struktury.

Osobiście, interesowałaby mnie jeszcze analiza oprogramowania w kontekście pracy emocjonalnej (a nie tylko pracy cyfrowej, która jest cennym elementem niniejszej dysertacji). Mogłoby to pozwolić na odczytanie software'u w szerszych kontekstach afektywnych, a tym samym w szerszych kontekstach jako takich.

Od strony formalnej, praca przygotowana jest z zachowaniem wszelkich wytycznych względem prac doktorskich i nie zawiera większych uchybień. Tekst zredagowany jest bardzo dokładnie i nie udało mi się w nim dopatrzeć usterek, z wyjątkiem drobnego błędu w przypisie na stronie 227, który bardziej niż błędem jest odstępstwem od przyjętego w pracy arkusza stylów dla bibliografii.

Na koniec, podzielę się z Autorką niezobowiązującym ale rozbudowanym pytaniem, które dotyczy głównych zagadnień pracy, a które formułuję za słynnym esejem Teda Nelsona pt. „The Right Way To Think about Software Design”. Czy istnieje właściwy sposób myślenia o oprogramowaniu oraz sposobach jego projektowania? Jeśli tak, to na czym się ono zasadza i co przewiduje w warunkach technologicznego kryzysu, o którym tak często słyszymy? Czy software zależy od projektowania czy też być może rozwijać się z własnej osobliwości, jeśli takowa istnieje? Ten ostatni człon pytania ma w sobie szczyptę intelektualnej prowokacji. Trudno jednak nie myśleć o softwarze w kontekście *nowacenu* (*Novacene*), zwłaszcza spekulatywnie.

Niezależnie od zaleceń, pytań i uwag, nie mam żadnych wątpliwości, że w przedstawionej pracy, Pani Magister Anna Paprzycka spełnia wszelkie wymogi stawiane rozprawom doktorskim. Niniejszym, z pełnym przekonaniem przyjmuję przedstawioną przez Doktorantkę pracę jako podstawę dopuszczenia jej do dalszych etapów postępowania doktorskiego.

Dr hab. Anna Malinowska, Prof. UŚ



Signed by /
Podpisano przez:

Anna Helena
Malinowska
Uniwersytet Śląski
w Katowicach

Date / Data:
2024-03-01 21:05

