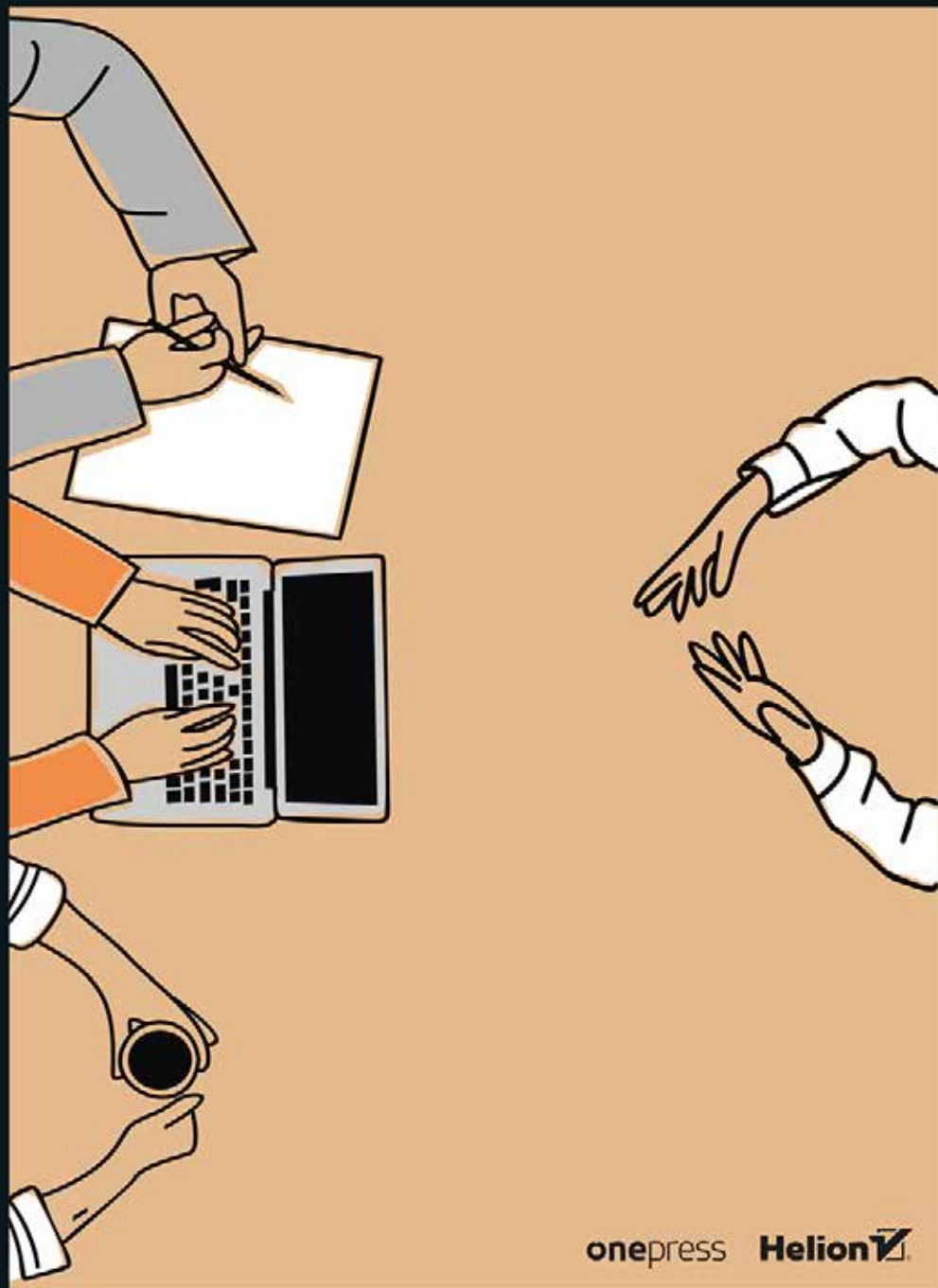


Rekrutacja w IT

Michał Kosiorek



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich.

Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Wojciech Ciuraj

Projekt okładki: Studio Gravite / Olsztyn

Obarek, Pokoński, Pazdijowski, Zaprucki

Grafika na okładce została wykorzystana za zgodą Shutterstock.com

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: onepress@onepress.pl

WWW: <https://onepress.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://onepress.pl/user/opinie/rekwit>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-289-0925-0

Copyright © Michał Kosiorek 2024

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

SPIS TREŚCI

WSTĘP _____	11
-------------	----

ROZDZIAŁ I

DLACZEGO WARTO DOBRZE ZASTANOWIĆ SIĘ NAD PODJĘCIEM PRACY JAKO REKRUTER _____	15
---	----

Rekruter — cyfrowy nomada _____	15
Ścieżka kariery rekrutera w IT _____	18
Rekrutacja w IT — dla kogo? _____	18
<i>Gdzie szukać pracy w rekrutacji w IT?</i> _____	20
<i>Talent sourcer</i> _____	21
<i>IT recruiter</i> _____	22
<i>Senior IT recruiter</i> _____	23
<i>Staffing manager</i> _____	23
<i>HRBP</i> _____	25
Rekrutacja — jak szukać klientów? _____	26
<i>Gdzie zacząć znajdowanie klientów?</i> _____	26
<i>No Fluff Jobs</i> _____	27

<i>justjoin.it</i>	28
<i>Rocket Jobs</i>	28
<i>LinkedIn</i>	29
<i>Jak nawiązać kontakt z klientem?</i>	29
<i>Jak napisać dobrego maila?</i>	33
<i>Szeroki networking</i>	34
<i>Pozyskiwanie klientów</i>	34
<i>Rozmowy z klientami</i>	34
<i>Konferencje</i>	35
Ile zarabiają rekruterzy? Analiza portali internetowych	39
<i>Stawki na LinkedIn</i>	40
<i>Stawki na Rocket Jobs</i>	41
<i>Stawki na No Fluff Jobs</i>	42

ROZDZIAŁ II

CZYM JEST OUTSOURCING W IT 45

Na czym polega outsourcing usług IT?	45
Rodzaje outsourcingu	47
Korzyści związane z outsourcingiem	48
Ile kosztuje outsourcing?	49

ROZDZIAŁ III

MINDSET REKRUTERA 51

Manifest Agile — czym jest i jak go stosować?	51
Czym jest metoda Agile?	51
Manifest Agile — założenia i wartości	52

Rozwój oprogramowania według Agile	54
Jak wdrożyć Agile w swojej firmie?	55
Tempo, inicjatywa — rozwlekły proces	56
Zrozumienie roli i wymagań — kto jest stakeholderem procesu rekrutacji	60
Self improvement, czyli zrozumienie roli dewelopera	61

ROZDZIAŁ IV

ZROZUMIEĆ TECH STACK 65

Back-end developer	65
<i>Bazy danych</i>	66
Front-end developer	67
<i>React.js developer</i>	67
<i>Vue.js developer</i>	69
<i>Svelte developer</i>	69
<i>Angular developer</i>	69
<i>TypeScript developer</i>	70
Full stack developer	70
Mobile app developer	71
<i>Jakie umiejętności powinien posiadać mobile app developer?</i>	72
Czym zajmuje się iOS developer?	72
<i>Umiejętności, które powinien posiadać iOS developer</i>	73
<i>Tester automatyzujący aplikacje iOS</i>	74
Czym zajmuje się Android developer?	75
<i>Tester automatyzujący aplikacje Androida</i>	76
Co to jest Flutter i kim jest Flutter developer?	78
<i>Kim jest Flutter developer?</i>	79

Czym zajmuje się React Native developer? _____	80
Czym zajmuje się cloud engineer? _____	81
<i>Początki chmury obliczeniowej</i> _____	82
<i>Rynek</i> _____	83
<i>Strefy dostępności Azure i AWS</i> _____	83
<i>Google Cloud Platform (GCP)</i> _____	84
Czym zajmuje się machine learning developer? _____	84
<i>SQL jako podstawa</i> _____	85
<i>Standard REST API i chmura</i> _____	85
Czym zajmuje się embedded developer? _____	86
<i>Czym właściwie zajmuje się embedded developer?</i> _____	86
Czym zajmuje się DevOps _____	87
<i>Główne założenia DevOps</i> _____	87
<i>Na czym polega praca na stanowisku inżyniera DevOps?</i> _____	88
Czym zajmuje się product designer? _____	88
<i>Dlaczego to takie ważne, aby strony były zaprojektowane zgodnie ze sztuką UX?</i> _____	89
Czym zajmuje się UX/UI designer? _____	89
Czym zajmuje się product owner? _____	90
<i>Jaka jest rola product ownera w Scrumie?</i> _____	90
<i>Na czym skupia się product owner?</i> _____	91
Czym zajmuje się scrum master? _____	92
<i>Jakie cechy powinien posiadać kandydat na scrum mastera?</i> _____	93
SaaS developer _____	93
SAS developer _____	94
MES developer _____	94
ERP developer _____	95
TMS developer _____	95

Deweloper systemów zarządzania magazynem (WMS)	97
Code review	98
Game developer	101
VR, AR i MR	104
Blockchain developer	108

ROZDZIAŁ V**PROJEKTY IT 111**

Rekruter nielubiany... Konkretny JD	119
<i>Must be, good enough</i>	120
<i>PSK — poznaj swojego klienta, poznaj projekt</i>	120
<i>Blind CV — po co?</i>	122

ROZDZIAŁ VI**TECH STACK I TOOLE REKRUTERA 125**

Narzędzia rekrutera	125
<i>ATS</i>	125
<i>Slack</i>	127
<i>Miro</i>	129
<i>Glip</i>	130
<i>Discord</i>	132
<i>Notion</i>	133
<i>Canva</i>	134
<i>Google Workspace</i>	135
<i>todoist</i>	137
<i>Zoom</i>	138

<i>Microsoft Teams</i> _____	140
<i>Google Hangouts</i> _____	141
<i>Trello</i> _____	143
LinkedIn Recruiter — słuszny wybór? _____	144
<i>Czym Jest LinkedIn Recruiter?</i> _____	144
<i>Dla kogo i jakie korzyści?</i> _____	145
<i>Czy LinkedIn Recruiter jest dla każdego?</i> _____	146
<i>Plusy i minusy korzystania</i> _____	146
<i>Jak działa wyszukiwarka LinkedIn Recruiter</i> _____	147
<i>Podsumowanie — wskocz na wyższy level rekrutacji!</i> _____	159
Boolean Search _____	160
<i>Boolean Search na LinkedIn!</i> _____	160
<i>Przykład w praktyce — poszukiwanie programisty Java w Warszawie</i> _____	162
<i>Boolean Search — złote narzędzie</i> _____	164
X-Ray Search — nowa era efektywnej rekrutacji _____	164
<i>Co to jest wyszukiwanie za pomocą X-Ray Search?</i> _____	164
<i>Zalety wyszukiwania X-Ray w rekrutacji</i> _____	165
<i>Jak wdrożyć wyszukiwanie X-Ray w procesie rekrutacji</i> _____	165
<i>Operatory w wyszukiwarce w praktyce</i> _____	166

ROZDZIAŁ VII

PRIORYTYZACJA ZADAŃ, TIMEBOXING SPOTKAŃ ___ 169

Zarządzanie czasem na podstawie timeboxingu _____	169
<i>Czym jest timeboxing?</i> _____	170
<i>Jak wykorzystać technikę Pomodoro w efektywnym planowaniu pracy?</i> _____	170

Podział dnia na strefy — <i>deep work</i> i spotkania	171
<i>Jak swoje dni planują najlepsi?</i>	172
Teoria już była, pora na praktykę	173
<i>Odnieśmy zarządzanie czasem do wątku programisty</i>	175
<i>Jak to działa — władanie czasem i priorytetami</i>	175
Candidate summary — experience	178
<i>Ocena kandydata w procesie rekrutacji</i>	178
<i>Czym jest podsumowanie kandydata?</i>	179
<i>Dlaczego podsumowanie jest istotne?</i>	179
<i>Jak przygotować dobre podsumowanie kandydata?</i>	180
Zalety live coding w rekrutacji IT	181
<i>Przykłady zastosowań live codingu w rekrutacji IT</i>	182
Employer branding w rekrutacji IT	183
<i>Co to jest employer branding?</i>	183
<i>Jak employer branding przerzuca się na rekrutację?</i>	184
<i>Jakie korzyści niesie employer branding?</i>	184
<i>Jak budować dobry wizerunek firmy w IT?</i>	185
<i>Znaczenie employer branding w procesie onboardingu kandydatów IT</i>	187

ROZDZIAŁ VIII

QUIET QUITTING W IT

— CZYM JEST I JAK MU PRZECIWDZIAŁAĆ? 191

Czym jest quiet quitting?	191
<i>Jak quiet quitting wpływa na branżę IT?</i>	192
<i>Jakie negatywne konsekwencje dla firmy niesie quiet quitting?</i>	193
<i>Jak przeciwdziałać quiet quittingowi w swojej firmie?</i>	193

ROZDZIAŁ IX**REKRUTACJA NA REKRUTERA — PRZYKŁAD _____ 195**

Przygotowanie się przed rozmową _____	196
Jak wygląda przykładowa rozmowa rekrutacyjna _____	198
<i>Etap 1.</i> _____	199
<i>Etap 2.</i> _____	199
<i>Etap 3.</i> _____	201
<i>Etap 4.</i> _____	202
<i>Etap 5.</i> _____	203
Przykładowe zadanie rekrutacyjne _____	203
<i>Przykładowe zadanie rekrutacyjne</i> _____	204
<i>RODO (ogólnie) w kilku punktach</i> _____	212

MINDSET REKRUTERA

MANIFEST AGILE

— CZYM JEST I JAK GO STOSOWAĆ?

Chociaż Agile nie jest niczym nowym, zasada zwinnego zarządzania projektami wciąż budzi w branży wiele kontrowersji. Wiele działów HR krytykuje ją z powodu zbyt dużego komplikowania zarządzania, z kolei inni twierdzą, że nic nie wspomaga pracy tak dobrze jak ona. Choć same założenia Agile'a mają już 20 lat, dyskusja o wdrażaniu tego modelu wciąż jest żywa. Czy sprawuje się on lepiej niż waterfall i czy warto wdrażać go w swojej firmie? Przekonajmy się!

CZYM JEST METODA AGILE?

Metoda Agile to metoda zwinnego zarządzania projektami oprogramowania. Opracowana została w 2001 roku przez 17 deweloperów chcących uczynić projektowanie bardziej lekkim i nastawionym na zmiany. Deweloperzy ci stworzyli **Manifest dla deweloperów programowania**,

zwany także **Manifestem Agile**. Zawiera on główne zasady, według których można zarządzać projektami w sposób zwinny — alternatywny wobec starych metod, takich jak model kaskadowy. Na podstawie tego modelu składają się cztery wartości oraz dwanaście zasad, które ustanawiają nowy sposób projektowania oprogramowania.

Kod tworzony według zasad Agile to kod prosty, często testowany oraz skupiony na dostarczaniu aplikacji w jej funkcjonalnych częściach. Projekt tworzony jest więc częściowo — inaczej niż w metodzie kaskadowej, w której każda jego część określana jest na samym początku procesu i dopiero po dostarczeniu wszystkich z nich uznać go można za kompletny. W Agile konstruowanie projektu rozwija się symultanicznie z jego postępowaniem — nie ma w nim skomplikowanych planów, jest jedynie proste planowanie kolejnych kroków, które pozwala procesowi tworzenia oprogramowania nabrać potrzebnej zdaniami wielu zwinności.

MANIFEST AGILE — ZAŁOŻENIA I WARTOŚCI

Siedemnastu twórców Manifestu Agile spotkało się w resorcie Snowbird w amerykańskim Utah, aby opracować najważniejsze wartości, którymi kierować powinno się zwinne zarządzanie projektami. Manifest ten opiera się na cenieniu bardziej:

- **Ludzi i interakcji** od procesów i narzędzi.
- **Działającego oprogramowania** od szczegółowej dokumentacji.
- **Współpracy z klientem** od negocjacji umów.
- **Reagowania na zmiany** od realizacji założonego planu.

Twórcy przyznają przy tym, że nie oznacza to tego, że wartości wypisane po prawej stronie nie są cenione. Są one po prostu mniej ważne niż te, które znajdują się po lewej stronie. Wartości te pokazują nam w dużym skrócie, czym będzie programowanie Agile. Projektowanie oprogramowania według tej metody opierać się będzie przede wszystkim na możliwości bieżącego wprowadzania zmian. Niczego nie planuje się tutaj trzy lata do przodu — produkt powstaje właściwie spontanicznie, przy ciągłej współpracy z klientem i istniejącej możliwości modyfikacji.

Przykłada się minimalną wartość do wszelkich formalności, bo wyłącznie spowalniają one proces tworzenia działającego oprogramowania — zamiast tworzenia wielu stron umów i całych ksiąg dokumentacji, których nigdy nikt nie przeczyta, lepiej priorytetyzować realne potrzeby klienta i to, z jakim oprogramowaniem chce on pracować. Wizja klienta może się w końcu zmieniać w trakcie procesu — nie wszystko da się zaplanować przed rozpoczęciem produkcji. Dlatego właśnie warto, aby kod pakowany był w formę funkcjonalnych i działających części — sprawi to, że oprogramowanie będzie mogło być dostarczane w częściach, których kolejne specyfikacje będą mogły być uwzględniane wraz z bieżącymi potrzebami klienta. Kolejną ważną kwestią jest prostota kodu — w metodzie Agile dobry kod to taki, którego zrozumienie nie wymaga przebijania się godzinami przez dokumentację. Zawsze lepiej jest postawić na prostotę, nawet jeśli nieco zmniejszy to optymalizację czy szybkość danego programu — pozwoli to łatwiej edytować kod, aby na bieżąco móc reagować na potrzeby klienta, które często się zmieniają. W końcu to właśnie klient jest nadrzędną wartością Agile — programowanie zwinne opiera się na dostosowaniu się do jego potrzeb.

ROZWÓJ OPROGRAMOWANIA WEDŁUG AGILE

Tworzenie oprogramowanie według metody Agile to nie tylko wartości — w manifeście znajduje się także dwanaście zasad, na których powinno się oprzeć rozwój oprogramowania. Przyjrzyjmy się im:

- Najwyższy priorytet ma dla nas zadowolenie klienta dzięki wczesnemu i ciągłemu wdrażaniu wartościowego oprogramowania.
- Bądźcie gotowi na zmiany wymagań nawet na późnym etapie jego rozwoju. Procesy zwinne wykorzystują zmiany dla zapewnienia klientowi konkurencyjności.
- Dostarczajcie funkcjonujące oprogramowanie często, w kilkutygodniowych lub kilkumiesięcznych odstępach. Im częściej, tym lepiej.
- Zespoły biznesowe i deweloperskie muszą ściśle ze sobą współpracować w codziennej pracy przez cały czas trwania projektu.
- Twórzcie projekty wokół zmotywowanych ludzi. Zapewnijcie im potrzebne środowisko oraz wsparcie i zaufajcie, że wykonają powierzone zadanie.
- Najbardziej efektywnym i wydajnym sposobem przekazywania informacji zespołowi deweloperskiemu i wewnątrz niego jest rozmowa twarzą w twarz.
- Działające oprogramowanie jest podstawową miarą postępu.
- Procesy zwinne umożliwiają zrównoważony rozwój. Sponsorzy, deweloperzy oraz użytkownicy powinni być w stanie utrzymywać równe tempo pracy.
- Ciągłe skupienie na technicznej doskonałości i dobrym projektowaniu zwiększa zwinność.

- Prostota — sztuka minimalizowania ilości koniecznej pracy — jest kluczowa.
- Najlepsze rozwiązania architektoniczne, wymagania i projekty pochodzą od samoorganizujących się zespołów.
- W regularnych odstępach czasu zespół analizuje możliwości poprawy swojej wydajności, a następnie dostraja i dostosowuje swoje działania do wyciągniętych wniosków.

Zasady te w znaczny sposób poszerzają wartości: zgadzając się z nimi, są ich swoistym rozszerzeniem. Powtarzają one, że w Agile najważniejsza jest praca z klientem — to w końcu jego zadowolenie jest celem wyprodukowanego przez zespół „softu”. Nie należy bać się zmian — nic nie stoi na przeszkodzie, aby poprzez modyfikację projektu zwiększyć jego funkcjonalność. Dlatego właśnie projektować powinno się tylko w krótkim dystansie — kod powinno dostarczać się w działających kawałkach i najszybciej, jak to tylko możliwe. Nie należy bawić się w nic, co ogranicza czas spędzony na projektowaniu oprogramowania — wszystkie informacje powinny być przekazywane bezpośrednio między zespołami w celu zapewnienia jak najlepszej efektywności samoorganizujących się zespołów.

JAK WDROŻYĆ AGILE W SWOJEJ FIRMIE?

Wprowadzenie modelu Agile do swojej firmy wciąż często uznawane jest za kontrowersyjne. Wobec modelu tego narosło wiele niepotrzebnych i krzywdzących mitów. Dlatego ważne jest podejście do tego tematu z otwartym umysłem i rzetelną wiedzą.

Przed wszystkim zastanów się, czy Twoja firma potrzebuje Agile — model ten najlepiej sprawdza się w organizacjach, które produkują

złożone systemy i produkty podlegające regularnym, nieliniowym zmianom. Stosowanie Agile przy produkcji prostego softu nie ma zbytniego sensu — najważniejszym elementem tej metody jest w końcu brak planowania, który powinien być raczej koniecznością niż wyborem.

Jeśli zdecydujesz, że tworzenie oprogramowania według Agile jest dobrym wyborem dla Twojej organizacji, warto zainwestować w odpowiednie szkolenia z zarządzania projektami w tej metodzie. Staraj się wybrać szkoleniowca z dużym doświadczeniem, które przynosić będzie realną wiedzę — na organizacji swojej firmy nie warto w końcu oszczędzać, gdyż to właśnie ona w dużej mierze odpowiedzialna jest za sukces przedsiębiorstwa.

Zwinne programowanie to niezwykle ciekawy model, który pozwala odmienić los wielu przedsiębiorstw — jest on jednak na tyle dobry, na ile sprawdzi się w konkretnych przypadkach. Nie zawsze wdrożenie Agile będzie idealnym sposobem na poprawienie efektywności firmy. Jeśli pracujesz jednak nad złożonymi systemami, które nieustannie się zmieniają, nie ma lepszej metody od Agile! Postaw na programowanie w zwinny sposób, dzięki czemu zwiększysz efektywność firmy przez ograniczenie zbędnych formalności i tworzenie konkretnych, działających części programu przy ciągłej współpracy z klientem mającej na celu zmaksymalizowanie jego zadowolenia.

TEMPO, INICJATYWA — ROZWLEKŁY PROCES

Zaczynamy temat... tutaj włożymy kij w mrowisko.

Z perspektywy rekrutera i HR to temat rzeka. Wszystko jest pięknie, jeżeli jest dość czasu na proces rekrutacyjny. Wtedy można faktycznie bez stresu podchodzić do opowiadania o procesie rekrutacyjnym

składającym się z jednego, dwóch czy czterech etapów. Pamiętajmy jednak, że każdy kij ma dwa końce. Czas ucieka, a ofert na stołach deweloperzy mają zazwyczaj kilka. Z jednej strony mamy zatem zespół deweloperski spragniony „procesu”, w którym w pierwszej iteracji odbywa się screening CV kandydatów i na tej podstawie zastanawiamy się wspólnie, jak ułożyć *pipeline* dostępnych kandydatów adekwatnych do danej roli. Nie ma znaczenia, czy używamy ATS (ang. *application tracking system*), czy też zakładamy wątek lub kanał rekrutacyjny na slacku, Discordzie czy Glipie. Cel jest jeden: poukładać kandydatów w odpowiedniej kolejności z zachowaniem ustalonych priorytetów.

Rozmawiamy zatem z kandydatem (wybaczcie, ale przyspieszam wątek, żeby mówić o *tech interview*). Więc rozpoczynamy pierwszą z na przykład sześciu rozmów. Okazuje się, że pozostałe pięć możemy zrealizować w kolejnym tygodniu (sytuacje w aktualnym projekcie nie pozwalają na udział w rozmowach, zdarzyły się urlopy czy nagłe nieobecności). Niedobrze... A zróbmy sobie automatyzację procesu i poprośmy kandydatów o rozwiązanie zadania technicznego asynchronicznie. Założmy, że kandydat wyrobi się z jego rozwiązaniem w 16 godzin. Pojawia się kolejne pytanie: kiedy kandydat znajdzie 16 godzin ciągiem na nasze zadanie, skoro w 90% nadal jest zatrudniony? Zapewne w któryś weekend. Albo też będzie „szył kod” każdego dnia po godzinach. Odeśle zatem zadanie w terminie do dwóch tygodni. Ciekawe, ile ofert od konkurencji otrzyma w tym czasie. Zapewne sporo. Wszystko jest pięknie, jeżeli rekrutujemy do Apple’a czy topowych firm, bo wtedy kandydatowi zależy, a my nie martwimy się zbytnio o czas z uwagi na to, że chętnych mamy całą kolejkę...

Niestety... prawda jest taka, że sytuacja idealna nie jest codziennością. My nie pracujemy jako rekruterzy dla Steve’a Jobsa (trzymam kciuki, w końcu *Sky’s the limit*), zatem jesteśmy po prostu klasyczną firmą, klasycznym software housem. Wyróżniamy się jedynie stawkami czy też własnym produktem bądź portfelem klientów, dla których pracuje

nasza firma. Jest też niecodzienna atmosfera, praca zdalna i milion fajnych benefitów. Czyli umówmy się — normalność... Co zatem wybiera deweloper, skoro ma kilka ofert na stole, lepszy „hajs” i nie chce mu się babrać w długie procesy?

Mamy zatem zespół deweloperski w naszej firmie, który albo jest zawałony robotą, albo nie ma czasu na proces rekrutacji, lub też (sytuacja idealna) team deweloperski ma czas na rozmowy rekrutacyjne z kandydatami.

Mamy przykład idealnej firmy, dla której kandydaci są w stanie zrobić sporo, są w stanie poczekać czy przejść długi proces rekrutacyjny...

W końcu mamy też — normalność — bezseną konkurencję i dewelopera, który szybko musi podjąć decyzję. Są i tacy, którzy nie chcą zmieniać zespołu czy swojego miejsca. Komunikat wygląda wtedy tak: „Na ten moment nie szukam zmiany, ale jak trafi się coś ciekawego, to może rozważyć propozycję...”.

Nie ma co czarować, tak wygląda nasze podwórko. Mniej założeń nie będzie, będą natomiast inne zależności, które co rusz dokładają ciekawych wniosków.

Pod koniec 2022 roku miałem przyjemność rozmawiać z jedną z agencji rekrutacyjnych, która realizowała dla mnie cele rekrutacyjne. Mieli fantastyczną zasadę. Klient (czyli firma zlecająca rekrutację) oraz kandydat mieli 72 godziny na przyjęcie oferty lub na decyzję o zatrudnieniu kandydata. I za to ich pokochałem... dlatego, że zawsze możemy znaleźć lepszego kandydata. Zawsze możemy odbyć kolejną rozmowę rekrutacyjną, zawsze możemy wysłać kolejne zadanie rekrutacyjne do kolejnego kandydata. Ale co, jeżeli ten, który był naprawdę OK, w oczekiwaniu na naszą odpowiedź (i procesowanie innych kandydatów z lejka) po prostu przyjmie ofertę innej firmy? I o to w tym wszystkim chodzi... o tempo i inicjatywę w procesie rekrutacji.

Złoty środek, którego szala przechyla się raz w kierunku goryczy po stracie dobrego dewelopera, a raz w kierunku euforii i prawie „mamy to!”. Rollercoaster emocji, smak sukcesu i rozczarowań. Tempo i inicjatywa...

Znam wiele przykładów takich scen, zostawię je jednak na opowieści na spotkanie na Zoomie. Zawsze jednak powinniśmy być świadomi, w jaki sposób definiujemy *skill set* kandydata, którego szukamy. Co określimy jako *must be*? Co określimy jako *good enough*? Co będzie absolutnym **no go** dla dalszego procesu tego kandydata?

Zaraz... STOP. Ale czy tylko to się liczy? Czy liczy się faktycznie wszystko szyte na miarę kandydata? Wydaje mi się, że nie do końca... **U nas też musi działać!**

Patrząc z perspektywy swojego zespołu, firmy, w której pracujemy jako wewnętrzni rekruterzy, czy też agencji rekrutacyjnej, powinniśmy mieć szerszy punkt widzenia. Ten człowiek musi pasować do zespołu... w tym wszystkim nie liczy się tylko ilość ofert na stole, fajne stawki czy dopasowanie oferty do kandydata.

Niedopasowany człowiek to ogromne konsekwencje:

- Frustracja zespołu deweloperskiego.
- Ryzyko niezadowolonego klienta (jego zespołu).
- Niedowiedzone „taski” podczas sprintów czy gorączka nieprzespanych nocy przed „releasem”, a na koniec wściekły zespół QA, który rozkłada ręce z bezsilności.
- Ryzyko, że kilka osób będzie musiało poświęcać czas na rozwiązywanie zaognionych sytuacji między ludźmi w „dew teamie”.
- Spadek motywacji i w końcu zaczynamy zabawę od początku w „jeden... dwa... trzy, szukam dewelopera!”.
- Straciliśmy czas, sporo kasy i finalny efekt jest marny.

Jak zatem ułożyć magiczny proces?

Temat na pewno poruszymy nieraz na wspólnym Zoomie, natomiast tutaj zastanówmy się, co ma największe znaczenie.

Zacznijmy od zrozumienia, co w trawie piszczy **i z kim mamy gadać**.

ZROZUMIENIE ROLI I WYMAGAŃ — KTO JEST STAKEHOLDEREM PROCESU REKRUTACJI

Czyli tysiące pytań do stakeholderów procesu rekrutacji (tak... wiemy, oni często tego nie lubią... często potrzeba dogrywki z team leaderem czy deweloperem odpowiedzialnym za interview naszych kandydatów...

Dlaczego używam określenia „stakeholder procesu rekrutacji?”

Tutaj powinniśmy zastanowić się, co jest powodem zainicjowania tego procesu. Dzięki temu zrozumiemy też naszego inicjatora zamieszania.

Jak wiemy, mogą być różne powody rekrutacji, zatem inicjator też może za każdym razem być inny (*Team leader, PM, Sales Team/Account/* internalowe powiększenie firmy — HR wewnętrzny w porozumieniu z „x”). Tymi powodami mogą być:

- *Scale up* zespołu deweloperskiego, nowy projekt, nowy zespół tworzony na potrzeby projektu.
- Zastąpienie członka zespołu deweloperskiego z uwagi na nieodpowiednie kompetencje, niedopasowanie, eskalujące problemy.
- Uzupełnienie zespołu z uwagi na to, że członek zespołu deweloperskiego złożył wypowiedzenie.

Jak widzicie, jest kilka przyczyn, które napędzają start procesu rekrutacyjnego i już w tym momencie otwierają drogę do fajnych i ciekawych pytań, dzięki którym możemy wspólnie z inicjatorem tego procesu wypracować wartościowe odpowiedzi, a te z kolei przedstawić kandydatom podczas spotkania i pokazać przy tym naszą postawę. Postawę profesjonalistów, którzy zrozumieli przyczynę tego zamieszania. Kolejno oczywiście dokładamy rozumienie zagadnień typowo projektowych.

Mało tego! Mamy już namiastkę procesu rekrutacji i poznajemy kolejne ważne osoby w naszym procesie... czyli zespół rekruterski, w którym zasiądą zapewne deweloperzy mający weryfikować techniczne umiejętności naszego przyszłego „unicorna” czy też „top playera”, którego wszyscy szukają.

Teraz zaczyna się tworzenie listy pytań do zlecającego proces rekrutacji. W ostatnim rozdziale znajdziesz listę bardzo przydatnych pytań, które zapewne dobierzesz do swojej palety zagadnień istotnych do omówienia przed startem procesu rekrutacji.

SELF IMPROVEMENT, CZYLI ZROZUMIENIE ROLI DEWELOPERA

Skup się! Będą smaczki...

Ten rozdział trochę zajmie... mam nadzieję, że stanie się on jednym z tych, do których często wrócisz.

Dlaczego zaczynam od zrozumienia roli dewelopera? Wyobraź sobie, że szukasz pracy **albo...** przypadek nr 2: praca szuka Ciebie i dostajesz wiadomość z ofertą. Załóżmy też, że chcesz kliknąć ofertę. Klikasz

i czytasz... nagle Twój mózg zaczyna przechodzić z systemu pierwszego (intuicja) na system drugi (racjonalne myślenie) i dostrzegasz w ofercie brak informacji...

- Brakuje stawki — widełek określających wynagrodzenie.
- Brakuje informacji o wielkości zespołu, do którego trafisz, czy o liczbie osób w firmie.
- Brakuje opisu problemów do rozwiązania, wyzwań, jakie będą na Ciebie czekały po przekroczeniu progu firmy.
- Brakuje informacji o procesie rekrutacji (ile rozmów będzie potrzebnych i z kim).

Widzisz za to ogromne uznanie i podziw za świetnie dopasowane umiejętności i doświadczenie opisane na Twoim koncie w LinkedIn i zaproszenie do rozmowy. Nie czujesz się oszukany(-na)? No pomyśl... chcesz obejrzeć film na Netfliksie, włączasz krótkie intro, czy czytasz opinie na temat tego, czy warto poświęcić czas na obejrzenie filmu?

Zrozum zatem dewelopera. Tak jak dla Ciebie, tak i dla niego czas jest bardzo istotny. Dlaczego wzajemnie mamy go tracić na niepewne kwestie?

Szanowanie czyjegoś czasu oraz świadomość dobrze kreowanych pytań ma ogromne znaczenie w branży HR. To czas jest najważniejszą walutą na świecie. Nie marnuj go zatem przez brak informacji czy brak konkretności.

Przykładowe pytania do zlecającego proces rekrutacji, czyli klienta.

- Jak wielki jest zespół?
- W jakich godzinach pracuje zespół?
- Jeżeli mamy *overlap* (inne strefy czasowe z klientem), to w jakich godzinach odbywają się spotkania zespołów np. z Polski i USA?

- Czy w „dew teamie” mamy QA (testerów)? — # *nie zawsze pytanie jest adekwatne do roli!*
- Dlaczego zaczynamy ten proces rekrutacji? Co jest przyczyną? Jaki problem powinien rozwiązać deweloper?
- Jakiego doświadczenia i „skilla” oczekujemy od kandydata?
- Ile procent jego zadań to zadania czysto deweloperskie, a ile np. testerskie? # *nie zawsze pytanie jest adekwatne do roli!*
- Jaką stawkę możemy oferować kandydatom, jakie widełki zakładamy?
- Jak wygląda kwestia sprzętu dla kandydata oraz kwestia wizyt w biurze?
- Czy kandydat będzie pracował tylko przy kodowaniu, czy też będzie musiał posiadać bardzo dobre kompetencje biznesowe (biznesowy performance), tak żeby samemu rozmawiać z klientem na partnerskim poziomie i „challengować” jego pomysły?
- Kiedy kandydat powinien wystartować w nowej roli?
- Kto będzie odpowiedzialny za transfer wiedzy, wdrożenie kandydata? Ile czasu zakładamy na *onboarding* kandydata?
- Jak długo będzie trwał projekt? Czy klient będzie rozwijał projekt (produkt) w przyszłości?
- W jakiej fazie znajduje się projekt (produkt) do którego szukamy kandydata? Czy kandydat będzie miał do czynienia z projektem kodowanym od zera, czyli od „green fieldu”, czy mowa o pracy przy refaktoryzacji starego, zastanego kodu?
- Jakie dodatkowe umiejętności powinien posiadać kandydat (bazy danych, platformy „cloudowe” AWS/GCP? # *nie zawsze pytanie jest adekwatne do roli!*

Oczywiście powyższe pytania to tylko kilka propozycji, natomiast każde z nich otwiera kolejne szuflady dające odpowiednie konteksty do dalszej eksploracji w celu określenia wymagań i oczekiwań.

Dlaczego w takim razie w tytule powyżej jest określenie *self improvement*? Dlatego, że do każdego „stacku” można dopasować kolejną serię pytań, choćby ze względu na wymagania projektowe, które szczegółowo pokazują nam, w jakim „stacku” czy „frameworkach” będziemy się poruszać.

Na myśli mam już kolejny krok... czyli omówienie, jakich języków (frameworków) dotyczy dana rola.

- Back-end.
- Front-end.
- Mobile app developer.
- Cloud.
- Full Stack.
- QA.

ltd.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJE

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

Najlepsza praca świata? Oczywiście rekruter w IT!

Branża IT jest, delikatnie mówiąc, nietypowa. Poza nielicznymi wyjątkami – jak specjaliści od sprzętu – większość osób realizujących projekty informatyczne, tworzących oprogramowanie i podtrzymujących działanie systemów może pracować zdalnie. Co więcej, zdalnie oznacza czasem operowanie z drugiego końca świata, w ramach międzynarodowych zespołów, których członkowie spotykają się ewentualnie *via* komunikatory i na zebraniach online. I wszystko to potrafi zadziwiająco sprawnie funkcjonować.

Skoro tak, skoro firmy z branży informatycznej są tak bardzo „zdalne”, może także pracownicy HR, może również rekruterzy byliby w stanie funkcjonować bez pojawiania się w biurze? Czytać maile na plaży na Bali, Zanzibarze czy choćby we Włoszech. Realizować powierzone sobie zadania w sposób uporządkowany i systematyczny, o godzinie, która najbardziej im odpowiada. I zarabiać przy tym kwoty rzędu kilkudziesięciu tysięcy na miesiąc. Jeśli taka wizja pracy w roli rekrutera w IT do Ciebie przemawia i teraz się zastanawiasz, jak zmienić wizję w rzeczywistość, to przestań marzyć, a zacznij czytać tę książkę. Jej autor konkretnie krok po kroku pokaże Ci, co musisz zrobić, by dołączyć do światowej armii *digital recruiter nomads* i pracować, skąd chcesz, jak chcesz i w godzinach, które Ci odpowiadają.



KOD KORZYŚCI
Sięgnij po więcej ▶



helion.pl



HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

ISBN 978-83-289-0925-0



Cena: 59,00 zł