Nie szkodzić

Wendell skrzywił się i podrapał po skroni. Wierzył, że uczenie się na błędach jest niezwykle ważne, szczególnie dla takich techników jak on. W końcu jego praca polegała na zapewnianiu innym ludziom możliwości korzystania z technologii w sposób płynny i przejrzysty. Zaczęło się od introspekcji, ustalenia, gdzie coś poszło nie tak i upewnienia się, że nie powtórzy się to ponownie. Ale czyje błędy? Jego? Lub innych? Wrócił myślami do spotkania zarządu zmiany, do pytań Henry'ego, do pośpiesznych odpowiedzi Gopala i do własnego, rosnącego poczucia niepokoju. Być może nie miał takiego doświadczenia, jakie mieli członkowie jego zespołu i wciąż uczył się o środowisku, ale nadal potrafił rozpoznać, kiedy sprawy nie układały się tak, jak powinny. „Więc chcę to naprawić…" Było już po godzinach pracy i wszyscy byli zdenerwowani. Imam siedział po drugiej stronie stołu i wyglądał na zniecierpliwionego. Tak naprawdę nie mógł wykonywać swojej pracy, dopóki Gopal nie skończył, więc spędzał czas na rozmowach i zadawania pytań – może było ich zbyt wiele. Gopal opuścił salę konferencyjną i poszedł pracować gdzie indziej, ponieważ wydawał się zirytowany uwagą Imama i nie mógł się skoncentrować. Wendellowi nie przeszkadzała obecność Imama i naprawdę podobało mu się, że był ktoś, kto znał aplikacje klientów na bieżąco podczas wdrażania zmiany. Wendell wziął głęboki oddech. „OK, myślę, że jestem gotowy". Imam skinął głową, wiercąc się na krześle. „Monitoruję aplikację, śmiało." Wendell dokładnie zapoznał się z instrukcjami i przepisał je w swoim terminalu. Nadal był nieco wstrząśnięty swoim poprzednim spotkaniem z Gopalem. Uprawnienia administratora były większym obciążeniem, ale nikt ich tak nie nazywał. Prawdopodobnie dlatego, że nikt nie pracowałby w ten sposób jako administrator systemu, pomyślał. Wdrożył nowy model; tym razem powinno pomóc w poprawnym powiązaniu adresów klientów i lokalizacji sklepów. Zmarszczył brwi. Model nie powiódł się z powodu błędu uprawnień. "Cholera." "Co to jest?" – zapytał Imam, chcąc wziąć udział. „Och, potrzebuję dodatkowego dostępu, aby model działał". Zdał sobie sprawę, że może dodać wspólne konto używane do wdrażania aplikacji do grupy klientów, aby uzyskać dostęp. Ale nie był pewien, czy to może zaszkodzić. „Jak myślisz, jak to najlepiej zrobić? Może powinniśmy zapytać właściciela bazy danych?" Imam zacisnął usta. „Prawdopodobnie poszli do domu. Daj mi pomyśleć. Wendell postanowił sprawdzić, gdzie jest Gopal. Otworzył komunikator i wpisał: „Jak idzie twoja praca?" „Uaktualniłem pierwszą wersję i schemat bazy danych" – odpowiedział Gopal po kilku chwilach. „Czy klient to sprawdził?" Wendell napisał. „W pobliżu nie ma nikogo, więc jutro". „W porządku, myślę, że możesz już promować nowy model" – powiedział Imam. „Posortowałem uprawnienia. Zmiana tymczasowa i wysłałem e-mail do właściciela bazy danych, aby o tym wiedział. Gdy sprawdzimy, czy model działa, nie będziemy go potrzebować". Wendell skinął głową. Kilka minut później Gopal wszedł do sali konferencyjnej. "Skończyłem." Imam uniósł brwi. "Już?" „Moja zmiana została zatwierdzona i potrzebujesz nowej wersji, więc kontynuowałem aktualizacje. Wszystkie bazy danych korzystają teraz z najnowszej wersji. Cóż, to chyba tyle. Idę do domu." „Do zobaczenia" – powiedział Wendell, a jego uwaga była rozdzielona pomiędzy czytaniem wyników nowego raportu i słuchaniem Gopala. „OK, w takim razie mogę przeforsować moją poprawkę" – powiedział Imam, łamiąc kostki. Wendell podniósł wzrok. Nie mógł się doczekać tego etapu ich pracy. Zmiana oprogramowania Imama miała na celu naprawę oprogramowania prognozującego, aby zapobiec jego bardzo powolnemu działaniu. Może to oznaczać znaczną poprawę jakości obsługi klienta. Imam pstryknął palcami. "Tak jak myślałem. Nasza zapora sieciowa powoduje całe spowolnienie. W tej nowej wersji zmieniłem port i teraz wszystko działa szybko." „Czy powiedziałeś zespołowi sieciowemu?" – zapytał Wendell. Imam wydawał się trochę rozproszony. „Nowa wersja bazy danych obsługuje niestandardowe skrypty, więc to też chcę sprawdzić." Wendell wiedział, że ta funkcja nie była jeszcze używana w środowisku produkcyjnym. Uważał, że Imam nie powinien testować dodatku i skupić się wyłącznie na naprawie spowolnienia. „No cóż, to nieoczekiwane" – powiedział Imam. „Wykorzystanie procesora jest ogromne i widzę, że czas wykonywania zapytań rośnie. Może to być silnik skryptowy. Wendell, czy

podczas debugowania możesz sprawdzić pełną funkcjonalność przewodnika? Naprawdę chcę już dziś wieczorem zakończyć zmiany. Zadzwonił telefon Imama. „Im mówi. Hej, Cezarze. Nie, nie dotykałem zapory sieciowej. Wendell i ja pracujemy nad naprawą powolności bazy danych. Tak, Gopal już poszedł do domu. Słuchaj, powinno być w porządku. Nie ma żadnego wpływu. Podniósł brodę. „Wendell, Caesar prześle ci logi zapory sieciowej. Czy możesz na nie spojrzeć? Wendell poczekał na wiadomość e-mail i zaczął czytać wpisy w pliku dziennika. Nie były to dane, z którymi pracował Imam, ale wydawało się, że większość z nich pochodziła z jednej konkretnej bazy danych i rzeczywiście zawierała informacje o lokalizacji klientów. „Myślę, że powinniśmy na razie zatrzymać bazę danych cusloc". Imam skrzywił się. „Jeśli to zrobimy, cała nasza praca pójdzie na marne". Wendell mimo wszystko rozważał kontynuowanie tej decyzji. „Zadzwonię do Jacoba" – dodał szybko Imam. „Może wiedzieć, dlaczego w bazie danych lokalizacji klientów występuje wysoki poziom aktywności zapory sieciowej". Znowu sięgnął po telefon. Wendell wskazał na biurko. „Włącz go na głośnik. Ja też chcę to usłyszeć. „Nie wiem" – powiedział Jacob, gdy wyjaśnili mu sytuację. „Naprawdę nie chcę zamykać produkcyjnej bazy danych. Zgadzam się jednak, że musimy zachować ostrożność i kontynuować badania. Klient może po prostu kierować wiele zapytań. Dajemy mu jeszcze pół godziny, a potem może uda nam się zrestartować bazę danych. W międzyczasie może poprosimy Cezara o zwiększenie poziomu pozyskiwania drewna. „Dziękuję" – powiedział Imam. Ponownie zadzwonił do Cezara. Wendell przeglądał podręcznik bazy danych. Wyglądało na to, że w niestandardowym silniku skryptowym nie było niczego, co mogłoby nagle spowodować ogromny wzrost wydajności procesora, zwłaszcza w przypadku niektórych podstawowych skryptów testowanych przez Imama. „No dobrze, Cezar mówi, że powinniśmy też zajrzeć do konsoli monitorującej. Być może dzięki temu lepiej zrozumiemy, dlaczego nagle widzimy taki duży ruch". Wendell uwierzytelnił swoją nazwę użytkownika w konsoli internetowej i poczekał, aż otworzy się panel monitorowania. Imam pochylił się. „Cóż, wygląda na to, że dane klientów są odczytywane i kopiowane przez zaporę sieciową". "Czy to normalne?" – zapytał Wendell. "Nie wiem. Klient mógł zmienić swoje zapytania. Powinniśmy zadzwonić pod telefon i sprawdzić, czy mogą pomóc w zlokalizowaniu problemu. „Powinniśmy powiadomić także Mike'a" – dodał Wendell. „Tak, powinniśmy to zrobić" – powiedział Imam i podniósł słuchawkę. „Hej, Mike. Imam. Przepraszam, że ci przeszkadzam. Słuchaj, Wendell i ja przeprowadzamy konserwację… Włączył głośnik. Wendell obserwował pulpit monitorujący, podczas gdy Imam wyjaśniał sytuację. Mike nie wydawał się zły ani zniecierpliwiony. Wendell niemal spodziewał się, że menedżer będzie zirytowany, że sprawy nie układają się dobrze. „…i chcemy zatrzymać bazę danych cusloc." Mike zamruczał. "W porządku. Daj temu pół godziny, a potem zrób to. Imam entuzjastycznie pokiwał głową. "Ok spoko. Dziękuję, Mike.

Na drodze szkody

Chociaż niektóre idiomy nie są dobrze tłumaczone, wyrażenie w języku innym niż angielski, które brzmi „ci, którzy nie pracują, nie popełniają błędów", ma dziwne zastosowanie w świecie IT. Każdy z nas na pewnym etapie swojej kariery zawodowej popełnił ten czy inny błąd w trakcie wykonywania swojej pracy. Czasami zdarzały się stosunkowo błahe wpadki, a jedynymi świadkami byli nasze indywidualne wyrzuty sumienia. Czasami były to duże błędy, które powodowały awarie. We wszystkich przypadkach zrobiliśmy coś, co uważaliśmy za słuszne, a co okazało się błędne. Ponieważ systemy komputerowe stają się coraz bardziej krytyczne w codziennym działaniu i funkcjonowaniu kluczowej infrastruktury, takiej jak elektrownie, szpitale, sieci ruchu i sieci komunikacyjne, rola personelu IT zmieniła się z opiekunów narzędzi i platform w strażników cywilizacji (można powiedzieć galaktykę, ale to by ją popchnęło). Niewielu ludzi ma natychmiastowe poczucie, że ich praca, odległa od faktycznego zastosowania technologii, może mieć tak dramatyczne konsekwencje i wpływ. Jest to także wymiar etyczny, na który ludzie niekoniecznie się zapisali, a na pewno nie zostali przeszkoleni w ramach swojej pracy w IT. A jednak błędy są nieuniknione. Z biegiem czasu prawdopodobieństwo, że coś pójdzie nie tak w dziale IT, zbliża się do jednego. To jest temat dość oczywisty – niektóre jego aspekty omawialiśmy

już w całej książce. Na przykład rozdział 2 (Szanuj prywatność) szczegółowo omawia konsekwencje utraty danych. O własności intelektualnej mówiliśmy w Rozdziale 4 (Nie kradnij własności intelektualnej). Ale nie omawialiśmy ludzkiej dynamiki wokół błędów. Ogólnie rzecz biorąc, nikt nie lubi się mylić – w błędzie – ani przyznawać się do błędów. Takie podejście może być postrzegane jako negatywne lub autodestrukcyjne i może mieć bezpośrednie konsekwencje dla zatrudnienia. Z kolei ludzie często przyjmują postawę obronną, gdy rzucane są im wyzwania (ze strony innych osób lub danych, które konsumują), co może jeszcze bardziej sprawić, że będą jeszcze mniej skłonni lub skłonni do introspekcji, co z kolei sprawi, że ludzie będą mniej skłonni do identyfikowania błędów w swojej pracy i postępowaniu. W ten sposób administratorzy systemów, programiści i technicy dokonują naruszeń zasad etycznych, nawet nie zdając sobie z tego sprawy. Gdy zagrożone jest życie ludzi, trzy lub cztery przeskoki do sieci, cena ego może być zbyt wysoka, aby ją zapłacić. Źle zaprojektowane systemy lub niejednoznaczne wyniki mogą powodować dalsze zamieszanie moralne. Być może najbardziej charakterystycznym przykładem źle zaprojektowanego systemu jest awaria elektrowni jądrowej Three-Mile Island1 w 1979 r. Niejednoznaczne odczyty w centrum sterowania stacji doprowadziły operatora do przekonania, że w reaktorze było za dużo wody chłodzącej, co doprowadziło do ręcznego obejście i zwolnienie ciśnienia pary. Niewielu z nas może postawić się w sytuacji tej osoby i w pełni zrozumieć skalę problemu, przed którym stanęła ta osoba w momencie wypadku, ale ilustruje to znaczenie kontroli podczas podejmowania decyzji, które muszą być obecne w naszych działaniach przez cały czas. Ale szanse są przeciwko nam. Jesteśmy ślepi na własne błędy. Jest uniwersalny. Wystarczy spojrzeć na napisany przez siebie e-mail (lub rozdział w książce), a zauważysz, że rzadziej niż korektor dostrzeżesz błędy gramatyczne i ortograficzne. Dotyczy to również prac technicznych. Jeśli piszesz skrypty, kodujesz oprogramowanie lub uruchamiasz polecenia podczas konserwacji, możesz zauważyć oczywiste błędy w procedurach lub logice dopiero po fakcie. Zjawisko to jest wzmacniane przez efekt potwierdzenia. Jako ludzie jesteśmy bardziej skłonni do ignorowania dowodów i faktów, które są sprzeczne z naszymi przekonaniami i założeniami, a przywiązujemy dodatkową wagę i wagę do tych, które je potwierdzają. Z etycznego punktu widzenia może się to skończyć odrzuceniem ciągów myśli, metod lub rozwiązań, które niekoniecznie pokrywają się z tym, co masz na myśli. Z tego samego powodu możesz także ignorować lub minimalizować problemy.

Nie walcz z błędami: (1) są nieuniknione, (2) jesteśmy ślepi na własną pracę, (3) używamy efektu potwierdzenia, aby zignorować sprzeczne dowody.

Świadomość jest kluczem

Zrozumienie, że błędy się zdarzają, jest niezwykle istotne. Stanowi to podstawę zasady zwanej projektowaniem pod kątem niepowodzeń (DFF), która zakłada, że wystąpią nieoczekiwane rezultaty. Zamiast unikać lub ignorować takie ewentualności, należy je mapować i traktować jako pewnik; i przeciwdziałając im, opracuj odpowiednie narzędzia umożliwiające im zapobieganie, łagodzenie i identyfikację.

• Zapobieganie – Jeżeli znany jest błąd lub awaria lub można ją obliczyć z dużym prawdopodobieństwem, systemy należy projektować w taki sposób, aby nie powodowały wystąpienia błędów lub awarii. Zazwyczaj systemy będą miały marginesy bezpieczeństwa i będą progi normalnego działania.

• Identyfikacja – Możliwe jest, że systemy lub ludzie odbiegają od oczekiwanych warunków pracy, niezależnie od tego, czy jest to problem w kodzie, czy nieaktualna dokumentacja, co może prowadzić do błędu. Powinien istnieć system monitorowania, zaprojektowany w oparciu o marginesy i progi bezpieczeństwa, który powinien ostrzegać w przypadku wystąpienia błędów, najlepiej przed spełnieniem warunków błędu.

• Łagodzenie – zapobieganie błędom, błędom lub awariom może nie być możliwe. W niektórych przypadkach nie będzie prostego, deterministycznego wzoru pozwalającego przewidzieć możliwe warunki błędu. Jeśli jednak zostaną prawidłowo zidentyfikowane, potencjalne szkody można złagodzić. Na przykład trudno jest przewidzieć z wyprzedzeniem, kiedy może nastąpić awaria dysku twardego. Jednak powstałe szkody można złagodzić poprzez użycie wielu kopii danych na kilku dyskach, zastosowanie technologii RAID itp.

W dalszej części omówimy, jak wdrożyć różne elementy polityki DFF. Skupmy się bardziej na pojęciu szkody.

Nie wszystkie szkody są równe

Istotne jest również zrozumienie, że wymiar etyczny rozciąga się jedynie na sytuacje, w których dochodzi do jakiejś formy interakcji międzyludzkiej i opiera się na poczuciu prawa na rzecz osób prowadzących pracę lub dokonujących zmiany.

• Celowe wyrządzenie krzywdy – jest to prawdopodobnie najmniej prawdopodobny przypadek. Ale najtrudniej jest też temu zapobiec i złagodzić skutki, zwłaszcza jeśli robi się to od wewnątrz środowiska. Wyrządzanie umyślnej krzywdy jest również bezpośrednio sprzeczne z przykazaniem zawartym w tym rozdziale.

• Przypadkowa szkoda – w większości przypadków szkoda jest wynikiem wypadku – niezamierzonego błędu, którego rezultatem jest inny od oczekiwanego rezultatu. Może to wynikać z wielu przyczyn, takich jak nieodpowiednie lub przestarzałe procedury, niewystarczające umiejętności operatora, założenia przyjęte pod presją czasu lub innymi ograniczeniami, decyzje podjęte na podstawie niewystarczających lub nieprawidłowych danych i nie tylko. Przypadkowa szkoda wynika z dobrych intencji, ale nie zwalnia jednostki z odpowiedzialności domeny. Należy pamiętać, że w niektórych przypadkach możesz zostać oskarżony o zaniedbanie lub umyślną szkodę, jeśli istnieje znaczna rozbieżność między oczekiwaną wiedzą a zrozumieniem systemów i Twojej pracy. W innych przypadkach, zwłaszcza gdy istnieją konsekwencje dla prywatności lub bezpieczeństwa, niektóre osoby mogą zostać pociągnięte do odpowiedzialności prawnej za znaczną szkodę. Na przykład dyrektorzy generalni firm zajmujących się oprogramowaniem, instytucji finansowych lub medycznych często ponoszą osobistą odpowiedzialność za naruszenia danych lub wypadki w swoich firmach.

• Błędy – w niektórych przypadkach mogą występować wady funkcjonalne oprogramowania i sprzętu, które spowodują szkody. Przykładowo awaria dysku może skutkować utratą danych, co może być szkodliwe dla użytkowników tych danych, jednak nie jest to zdarzenie, które można przypisać działaniu jakiejkolwiek jednostki.

Wymiar etyczny istnieje przede wszystkim w tej drugiej kategorii. Zmiany systemowe, takie jak zabezpieczenia obwodowe i poprawki oprogramowania, mogą zaradzić celowym szkodom i błędom w kodzie, ale nie są w stanie poradzić sobie z logicznymi wadami naszego myślenia i postrzegania środowiska ani naszym poczuciem, że mamy rację.

Nie szkodzić

W środowisku IT większość ludzi będzie kojarzyć szkodę z wymierną szkodą spowodowaną bezpośrednio (i pośrednio) czyjąś pracą. Dotyczy to jednak niemal każdego aspektu interakcji i użytkowania systemów. W przypadku poważnego incydentu mogą wystąpić straty w postaci sekund straconych w oczekiwaniu na załadowanie aplikacji, co może skutkować wielomiliardowymi stratami.

Zanim będzie można naprawić szkodę, należy ją prawidłowo zidentyfikować, a zaczyna się to od każdego działania. Na zadania IT należy patrzeć przez pryzmat konsekwencji i podobnie jak w rozdziale 8 (Komunikowanie zmian) można je traktować jako eksperymenty naukowe. Możesz zadać sobie pytanie, jaki jest oczekiwany wynik działania, które wykonujesz? Co zrobisz, jeśli coś pójdzie nie tak? Czy rozważałeś sytuacje i scenariusze, które odbiegają od pożądanych rezultatów i czy wiesz, jak sobie poradzić z każdym z nich? Należy także pamiętać, że krzywda nie musi być skierowana w stronę innych (i tym samym wymagać komunikacji). Możesz sobie zrobić krzywdę! Na przykład możesz stracić cenną pracę, jeśli nie będziesz przechowywać odpowiednich kopii zapasowych. Jeśli używasz nieodpowiednich lub przestarzałych narzędzi i skryptów, być może będziesz musiał poświęcić więcej czasu na wykonywanie zadań administracyjnych systemu. Ponieważ temat jest tak szeroki i ogólny, bardzo łatwo go pominąć lub ukryć pod zasłoną codziennych procesów. Można jednak wypracować etyczne podejście do pracy skupione na krzywdzie i szkodach oraz wykorzystać narzędzia i metody, których nauczyliśmy się w poprzednich rozdziałach, ze szczególnym naciskiem na komunikację. Jeśli istnieje prawdopodobieństwo, że wyrządzisz krzywdę, zatrzymaj się i poinformuj. Zadania związane z pracą muszą być wymierne, abyś mógł zrozumieć poziom możliwych szkód, które mogą wyniknąć z nieprawidłowo wykonanych planów. Zazwyczaj ryzyko i szkoda są ze sobą ściśle powiązane, a w idealnym przypadku istnieje ich mapowanie 1:1; jeśli proponowana praca wiąże się z wysokim ryzykiem, potencjalny poziom szkód również będzie wysoki. Co ważniejsze, administratorzy systemów, technicy i inżynierowie muszą mieć świadomość, że ich praca może wywołać łańcuch niepożądanych rezultatów. Czasami jednak po prostu nie jest możliwe ani wskazane oczekiwanie, że każdy pracownik będzie pamiętał o pełnym zakresie możliwych zagrożeń i pułapek w środowisku w danym momencie. W tym celu należy również precyzyjnie określić prawdopodobieństwo wystąpienia szkody, aby inżynierowie wiedzieli, jak postępować i kiedy informować innych o swojej pracy. Rzeczywiście, jeśli konsekwencje są dobrze określone, ludziom łatwiej je zapamiętać i szanować. Wreszcie najbardziej krytycznym elementem w tym równaniu jest umiejętność zatrzymania się i cofnięcia o krok. Presja czasu i zmęczenie mogą zaburzyć naszą ocenę sytuacji, a administratorzy systemów często muszą pracować pod prąd, bez przekonania wpisując odpowiedzi swoim menedżerom żądającym odpowiedzi. Wszyscy znaleźliśmy się w sytuacji, w której przekroczyliśmy termin, zrzędliwi i zmęczeni, walcząc ze środowiskiem i zalewem pytań. Prawie zbyt łatwo jest nacisnąć przycisk Enter i pozwolić, aby coś się działo. W takich sytuacjach jeszcze bardziej konieczne jest zachowanie ostrożności. Chociaż wyrażenie „jeśli istnieją wątpliwości, nie ma wątpliwości” brzmi idealistycznie (lub banalnie), jest ono istotną częścią etycznego postępowania, szczególnie w przypadku osób na uprzywilejowanych stanowiskach, takich jak administratorzy systemów. Ale nie chcesz być osobą, która wstrzymała coroczną aktualizację, ponieważ nie miałeś pewności co do polecenia, tak samo jak nie chcesz być osobą, która spowodowała masową awarię. Nadmierne unikanie ryzyka w połączeniu ze swobodnym podejściem do obwiniania „małego faceta”, gdy coś pójdzie nie tak, wypacza nasz osąd i popycha nas do podejmowania pochopnych, nieetycznych wyborów. Jednym z moich pierwszych błędów jako menedżera było nakłonienie administratora pamięci masowej, aby kontynuował aktualizację serwera pamięci masowej, gdy wystąpiły problemy. Była sobotnia noc, a on powiedział mi: „Nie jestem pewien, co jest nie tak, ale mogę zacząć od nowa i naprawić to i zaktualizować w czasie przestoju, albo mogę się wycofać i spróbować innego dnia”. Oczywiście powiedziałem: „Zajmij się tym teraz”. Cóż… poprawka i aktualizacja nie zadziałały zgodnie z oczekiwaniami, a okno przestoju przekroczyło granicę, która trwała do niedzieli. Następnie większość tygodnia spędziliśmy na przywracaniu utraconych danych. Administrator magazynu postąpił słusznie; zatrzymał się i poinformował swojego menadżera. Jako menadżer byłem po prostu niecierpliwy. Kiedy otrzymasz zgodę na dokonanie zmiany w określonym czasie, nie oznacza to, że zrobisz to lub zginiesz. Jeśli możesz „umrzeć”, przestań to robić! Przeżyjesz, by spróbować kolejnego dnia. Na szczęście dostałem drugą szansę, gdy dowiedziałem się, czego nie robić następnym razem. Możliwe jest przyjęcie praktycznego podejścia „zatrzymaj i

poinformuj”; obejmuje także terminową komunikację, prawidłowe wykorzystanie danych i czynniki ryzyka. Przyjrzyjmy się, co zrobił Wendell i jego współpracownicy.

Nie wchodź sam

„Jak myślisz, jak to najlepiej zrobić? Może powinniśmy zapytać właściciela bazy danych?”

Dowiedzieliśmy się, jak ważna jest pisemna dokumentacja i ścisłe instrukcje pracy – chodzi o to, aby jak najmniej odbiegać od znanych wyników i móc szybko wrócić do stanu kontrolowanego. Wendell rozumie, że dodatkowy dostęp do bazy danych nie jest przedmiotem dyskusji, a może mieć negatywny wpływ na klienta. Zamiast kontynuować swoją pracę, zachowuje ostrożność i konsultuje się z imamem. Ogólnie rzecz biorąc, pozwolenie innym osobom na „kwestionowanie” Twojej pracy to dobry sposób na odfiltrowanie oczywistych problemów i błędów, których Ty jako autor możesz nie dostrzec. Jeśli potrafisz skutecznie „obronić” wybrane przez siebie narzędzia i metody, prawdopodobieństwo wystąpienia błędów jest mniejsze.

„Naprawdę nie chcę zamykać produkcyjnej bazy danych. Zgadzam się jednak, że musimy zachować ostrożność i kontynuować badania”.

Jacob zachowuje ostrożność i jest wrażliwy na potencjalny wpływ, jaki takie posunięcie może mieć na klientów. Ale nie dlatego, że ma awersję do ryzyka. W rzeczywistości przyznaje, że istnieje problem i chce dokładniej go zbadać. Dodatkowe informacje mogą pomóc administratorom systemów i technikom w podejmowaniu bezpieczniejszych i bardziej etycznych decyzji. – Powinniśmy też powiadomić Mike’a. Praktycznie każda organizacja lub grupa ludzi dzieląca jakąś sprawę tworzy hierarchię władzy, oficjalnie lub nieoficjalnie. Jest to rzecz naturalna, dlatego rola menedżera istnieje od zarania cywilizacji. Menedżerowie muszą między innymi podejmować decyzje. Rzeczywiście, integralną częścią obowiązków kierowniczych jest czasami wzięcie odpowiedzialności za problemy i sytuacje, które niekoniecznie mają dobrze określony wynik. Wydaje się, że problem z bazą danych narasta poza możliwościami i uprawnieniami członków zespołu zaangażowanych w konserwację. W tym momencie istnieje potrzeba wkroczenia kogoś, przeanalizowania problemu na bardziej ogólnym poziomie i podjęcia decyzji o dalszym sposobie działania. Zapobiegnie to podejmowaniu przez członków zespołu nieetycznych kroków. Mike nadal stanie przed dylematem, czy nie wyrządzić krzywdy, ale jego punkt widzenia, przeszkolenie i uprawnienia będą inne niż w przypadku programistów i administratorów systemu.

Zwolnij i skonsultuj się

„Moja zmiana została zatwierdzona i potrzebujesz nowej wersji, więc kontynuowałem aktualizacje. Wszystkie bazy danych korzystają teraz z najnowszej wersji.”

Jednostronna decyzja Gopala o dokończeniu całej pracy bez skoordynowania jej z resztą zespołu stanowi naruszenie przykazania. Fakt, że jego zmiana została zatwierdzona, nie upoważnia do bezmyślnej pracy. Powinien był zatrzymać się po zakończeniu pierwszej aktualizacji systemu i poczekać, aż Imam i Wendell przetestują nowy algorytm. Jeśli wystąpił problem z aktualizacją bazy danych, problem ten został wielokrotnie powtórzony w innych systemach i zamiast dotyczyć pojedynczego hosta, problemy z aktualizacją mogłyby teraz dotyczyć całej bazy produkcyjnej. Wendell mimo wszystko rozważał kontynuowanie tej decyzji. Myśli Wendella rezonują z nami wszystkimi. Wszyscy znajdowaliśmy się w sytuacjach, w których liczył się czas i musieliśmy działać szybko, być może nawet bez uwzględnienia wszystkich faktów i implikacji. W rzeczywistości działanie pochopne lub wbrew ustalonym zasadom zwykle kończy się większymi problemami i większymi szkodami. Jako ludzie zwykle skupiamy się na cudownych ratunkach i strzałach jednego na milion, ale są to odosobnione

przypadki czystego szczęścia. Teraz Wendell najprawdopodobniej ma rację w swoich założeniach, powinien jednak upewnić się, że swoim czynem nie spowoduje jeszcze większego problemu. Oznacza to możliwość pełnego uzasadnienia i zmapowania skutków zatrzymania produkcyjnej bazy danych klienta. W idealnym przypadku takie scenariusze będą częścią polityki DFF i uwzględnione w planach zmian i konserwacji. Ponieważ Wendell nie ma wszystkich informacji, rozsądniej jest nic nie robić.

Jeśli przypadkowo wyrządziłeś krzywdę, powiedz

Błędy są nieuniknione. W jakiś sposób nastąpi sytuacja, w której Twoje działania spowodują szkody wewnątrz środowiska IT. Ta nieuchronność jest częścią ryzyka zawodowego w miejscach pracy, w których występuje intensywny rozwój technologii i inżynierii, i powinna być nierozerwalnie związana z Twoją odpowiedzialnością na swoim stanowisku. Kiedy zdarzy się krzywda, Twoje zachowanie po fakcie będzie miało ogromny wpływ na konsekwencje. Jeśli zignorujesz problem lub odsuniesz go na bok, albo, co gorsza, spróbujesz go zatuszować, ludzie mogą przypisać Twoim działaniom złośliwość i celowość. Niewinny błąd można zinterpretować jako umyślną krzywdę i chociaż możesz stracić pracę z powodu pracy, która nie jest zadowalająca, na pewno ją stracisz, jeśli zostaniesz przyłapany na próbie zatarcia śladów swoich błędów, a nawet możesz zostać pociągnięty do odpowiedzialności karnej naładowany. Ogólnie rzecz biorąc, ludzie są znacznie bardziej tolerancyjni wobec tych, którzy przyznają się do błędów i biorą odpowiedzialność za swoją pracę. To bardzo ludzka rzecz: daje im poczucie zaufania i pewność, że postępujesz uczciwie – i że nie będziesz próbował obwiniać ich za swoje błędy! Co więcej, twoi rówieśnicy i współpracownicy również zrozumieją, że pewnego dnia mogą wyrządzić krzywdę, więc w ramach waszej wspólnej sprawy będzie pewien poziom solidarności. Środowisko zaufania pozwala na znacznie lepszą współpracę, co może pomóc zminimalizować szkody i zmniejszyć ryzyko przyszłych błędów o podobnym charakterze. Wreszcie, im szybciej podejmiesz działania po wyrządzeniu szkody, tym większe są szanse na zatrzymanie wszelkich kaskadowych szkód, zmniejszenie kosztów i rozmiaru szkód oraz zebranie dzienników danych, które mogą pomóc w analizie sytuacji i poprawie odporności Środowisko IT. Dość często wszystkie historie naruszeń danych, o których słyszeliśmy, były zgłaszane miesiące lub lata po wystąpieniu naruszenia. Można sobie wyobrazić administratora systemu ignorującego alerty, pomijającego audyty, ukrywającego błąd, próbującego naprawić problem lub twierdzącego, że to nie jest wielka sprawa – zanim ktoś inny w końcu eskaluje problem. Następnie możesz sobie wyobrazić menedżera, który naciska na swój zespół, aby zminimalizował problem, znajduje winnego, wytyka błędy, a następnie zgłasza problem tak delikatnie, jak to możliwe. Z pewnością możesz sobie wyobrazić firmę próbującą znaleźć sposób na potraktowanie problemu jako drobnego, ograniczenie zakresu raportowania, zidentyfikowanie jednego winnego pracownika niskiego szczebla, a następnie zbieganie się z ogłoszeniem w ważnym dniu informacyjnym. I oczywiście, jeśli prawnicy zdobędą informację, zaczną żądać, aby sprawa była poważnym problemem, aby zakresem obejmował każdego, kto kiedykolwiek miał kontakt z systemem, a winni są menedżerowie najwyższego szczebla, którzy powinni podać się do dymisji i że każdy powinien otrzymać odszkodowanie. Ile miało miejsce naruszeń, gdy administrator systemu lub menedżer zatuszował problem? Ile zostało zignorowanych lub zmiażdżonych przez dyrektorów korporacji? Ilu przeszło przez prawników bez żadnego szumu prasowego? Świat nigdy się nie dowie…

Spotkałem się z sytuacjami administratora systemu, w których osoba, która spowodowała problem, „nie pamiętała", że cokolwiek robiła, podała jedynie niejasne szczegóły lub milczała, gdy czuła, że jest obwiniana. Z powodu takiego podejścia nigdy nie udało się zidentyfikować pierwotnej przyczyny problemu; procesów nie dało się ulepszyć i nie wprowadzono automatyzacji. W większości przypadków było jeszcze gorzej – wdrożyliśmy automatyzację i procesy, które obejmowały każdy możliwy scenariusz, który mógł spowodować problem. Spowodowało to stratę czasu i pieniędzy – pochłonięcie

cennych zasobów w celu wdrożenia i utrzymania rozwiązań. Najprawdopodobniej żadne z rozwiązań typu fishnet nie rozwiązałoby problemu.

Twój zespół chce pomóc

„Wykorzystanie procesora jest ogromne i widzę, że czas wykonywania zapytań rośnie. Może to być silnik skryptowy. Wendell, czy możesz sprawdzić pełną funkcjonalność przewodnika, podczas gdy ja będę go debugować?"

Imam uważa, że mógł wyrządzić krzywdę – a przynajmniej na tym etapie występują komplikacje w konserwacji, które mogą przerodzić się w poważny problem. Nie ukrywa tego i prosi Wendella o pomoc. Jest to pomocne na kilku poziomach. Po pierwsze, działanie Imama zwiększa widoczność potencjalnego problemu i zmniejsza ryzyko, że Wendell będzie kontynuował pracę, która mogłaby spowodować dalsze szkody. Po drugie, Wendell może być w stanie zapewnić użyteczną wiedzę i wiedzę specjalistyczną w stresującym momencie, co może prowadzić do szybszego i skuteczniejszego rozwiązania problemu. „Im mówi. Hej, Cezarze. Nie, nie dotykałem zapory sieciowej… Wendell, Cezar prześle ci kilka logów zapory. Czy możesz na nie spojrzeć? Decyzja Cezara o skontaktowaniu się z Imamem jest rozsądna. Chociaż Cezar sam nie wykonywał żadnej pracy związanej z konserwacją po pracy, niezależnie od tego czuje się odpowiedzialny za swoją dziedzinę pracy i decyduje się informować swoich kolegów o nietypowym ruchu w sieci. Udostępnia także dodatkowe dane, które mogą pomóc w rozwiązaniu problemu i tym samym zapobiegnięciu potencjalnym szkodom.

Nie ignoruj problemów

„Wendell, czy możesz sprawdzić pełną funkcjonalność w przewodniku, podczas gdy ja będę to debugować? Naprawdę chcę już dziś wieczorem zakończyć zmiany.

Dotykaliśmy tego przykładu wcześniej. Działanie Imama było krokiem we właściwym kierunku, ale nie zrobił on wszystkiego, co powinien był zrobić w tej sytuacji. Imam powinien był także powiadomić pozostałych członków zespołu zaangażowanych w prace po godzinach pracy, np. Caesara, aby nie doszło do zakłóceń w komunikacji i żadna praca nie była wykonywana w odosobnieniu, co pogłębiłoby problem i jeszcze bardziej utrudniło jego rozwiązanie. „Hej, Cezarze. Nie, nie dotykałem zapory sieciowej. Wendell i ja pracujemy nad naprawą powolności bazy danych. Tak, Gopal już poszedł do domu. Słuchaj, powinno być w porządku. Nie ma żadnego wpływu. Rozmowa telefoniczna Imama z Cezarem jest zdecydowanie naruszeniem etyki. Chociaż nie zainicjował rozmowy telefonicznej ze swoim kolegą, nadal miał możliwość zasygnalizowania Cezarowi potencjalnego problemu. Zamiast tego założył, że nie ma żadnego wpływu, czego ani on, ani Wendell nie ustalili. Jest to nieprawdziwa informacja, która może dać Cezarowi (lub innym członkom zespołu) fałszywe wyobrażenie o sytuacji. Jest oczywiste, że Cezar jest zaniepokojony, ponieważ zauważył anomalię w ruchu na zaporze ogniowej. Na tym etapie Imam powinien był koordynować pracę wszystkich uczestników i upewnić się, że wszyscy zostali poinformowani o konserwacji, problemie i wszelkich innych aspektach wydarzenia.

Zapobiegaj, identyfikuj i łagodź szkody

Można powiedzieć, że dobrze zarządzane środowiska IT są projektowane i architekturą pesymisty. Rzeczywiście właściwym podejściem do tworzenia solidnych i odpornych konfiguracji IT jest stosowanie zasady projektowania uwzględniającego awarie. Błędy są nieuniknione, w związku z czym wystąpią szkody. Zamiast tego unikać, właściwym podejściem jest sformułowanie problemu, określenie ilościowe, a następnie zastosowanie środków naprawczych jako integralnej części rozwiązania na każdym poziomie. Skrajnym przykładem jest Netflix, który wykorzystuje Chaos Monkey2, zaprojektowaną do celowego sabotowania części środowiska w celu przetestowania

odporności i redundancji systemów i usług. W większości przypadków taki poziom pozytywnego rygoru destrukcyjnego nie będzie konieczny, ale szkoda powinna być częścią architektury i procedur operacyjnych w środowisku IT. Możliwe jest ustrukturyzowanie architektury Projektowanie pod kątem awarii w postaci macierzy 3 × 3. Kolumny obejmują różne kategorie zarządzania szkodami – zapobieganie, identyfikacja i łagodzenie. Wiersze dotyczą różnych rodzajów szkód – umyślnych, przypadkowych i błędów. Pokazano to w tabeli

	Prevention	**Identification**	**Mitigation**
Deliberate harm	Physical security measures Network security measures Access control Permissions	Intrusion detection systems Active real-time monitoring of systems with well-defined thresholds and safety margins	Unknown
Accidental harm	Standard and privileged account separation Well-defined policies and procedures Change control Point of no return	Monitoring	Systems redundancy Data backups Backout plan Call for help
Bugs	QA and validation of tools and software	Monitoring	Systems redundancy Vendor support and/or use of supported software

Należy zauważyć, że łagodzenie skutków umyślnej szkody jest bardzo trudne, dlatego firmy i organizacje wymagają odpowiednich środków, aby zapobiegać wszelkim możliwym przypadkom umyślnej krzywdy i szybko je identyfikować. Nie zawsze może to być wykonalne, szczególnie jeśli winowajcą jest pracownik posiadający wewnętrzną wiedzę o systemach. Z drugiej strony zapobieganie szkodom zewnętrznym jest łatwiejsze, ale wymaga znacznych zasobów. Przypadkowe uszkodzenie obejmuje praktycznie każde działanie w świecie IT, dlatego też konkretne metody będą się różnić w zależności od scenariusza. Jednak nadal możliwe jest zastosowanie kilku ogólnych, uniwersalnych zasad, które mogą pomóc zminimalizować ryzyko przypadkowych obrażeń. Stosowanie separacji kont jest doskonałym przykładem świadomości, że mając konto uprzywilejowane można wyrządzić większe szkody i że należy minimalizować jego użycie. Dobrze zdefiniowane, aktualne procedury zmniejszają niejednoznaczność i minimalizują ryzyko źle wykonanego zadania utrzymaniowego. Kontrola zmian to świetna okazja do omówienia pracy ze współpracownikami i odkrycia luk i problemów w zaplanowanych działaniach.

Punkt bez powrotu i plan wycofania się

Redukcja szkód przypadkowych ma także wymiar czasowy. Praca w przestrzeni IT jest bardzo dynamiczna i często stresująca, a ludzie mogą popełniać błędy, nawet jeśli wszystko zostało poprawnie zaprojektowane. Żaden plan nie przetrwa pierwszego kontaktu z wrogiem. —Helmuth von Moltke

W tym przypadku żywym i oddychającym środowiskiem IT jest Twoim wrogiem. Jest to szczególnie prawdziwe, jeśli plany pracy wykraczają poza harmonogram, co często zdarza się w przypadku konserwacji systemu. Kiedy tak się dzieje, prawdopodobieństwo błędów i wpadek rośnie wykładniczo.

Może się zdarzyć, że Twój menadżer lub niecierpliwi klienci będą Ci dech w piersiach lub napięty termin, w którym musisz ukończyć pracę, a pracując pod czasem, możesz pominąć lub pominąć kluczowe kroki, które mogą prowadzić do szkód. Dzieje się tak, ponieważ zadania są często nieograniczone i nie mają jasno określonego planu odzyskiwania. Aby uniknąć tych niepewnych sytuacji, najlepiej założyć, że coś pójdzie nie tak i wykroczy poza harmonogram. Oznacza to, że każdy plan zmian powinien również uwzględniać punkt bez powrotu i pełny plan wycofania się, szczególnie w przypadku najważniejszych i krytycznych zmian. Punkt bez powrotu to etap w sekwencji pracy, w którym, jeśli sprawy nie układają się dobrze, nadszedł czas na wdrożenie planu wycofania się. Takie podejście pozwala na kontrolowany powrót do znanych warunków środowiskowych. Choć może to być zawstydzające lub frustrujące, wiąże się to z mniejszym ryzykiem niż ślepe podążanie naprzód. To właśnie wtedy dochodzi do katastrof.

Przełknij swoją dumę

Kolejnym istotnym elementem metodologii DFF jest jak najwcześniejsze komunikowanie problemów. Wszyscy mamy tendencję do kulewania się, gdy rozwiązujemy problemy. Jest to rzecz naturalna, a czasem bezpośrednie, osobiste wyzwanie dla naszej dociekliwej natury jako inżynierów i techników. Jednak wprowadzenie ważnych zmian w środowisku może być ryzykowne i może prowadzić do znacznych szkód. Informowanie innych o problemach daje możliwość skorzystania z zasobów i poproszenia o pomoc. Jeśli zatuszujesz problem, możesz po prostu zagłębić się w problem głębiej. Czasami najlepszą rzeczą jest przełknąć swoją dumę i postępować etycznie. W naszej drugiej historii z okopów IT rozmawialiśmy o „amnezji" następującej po incydentach. Taka postawa jest wysoce szkodliwa dla środowiska i ludzi, daleko wykraczająca poza rzeczywistą szkodę wyrządzoną w konkretnych przypadkach. Kiedy nie ma wystarczających danych, aby właściwie przeanalizować problemy, często niemożliwe jest zidentyfikowanie pierwotnej przyczyny. Oznacza to, że rozwiązania i środki zaradcze są stosowane bez jasnego zrozumienia problemów. Zespoły mają tendencję do przyjmowania pozycji defensywnej i korzystania z rozwiązań ogólnych, takich jak automatyzacja i procesy, które obejmują każdy możliwy scenariusz problemu, marnując czas i pieniądze oraz zabierając cenne zasoby na wdrożenie i utrzymanie tych rozwiązań. Na dłuższą metę może to być bardzo demoralizujące

Czas ma znaczenie

Wiemy już, że jeśli odkryjesz krzywdę, powinieneś o tym powiedzieć. Ale jest też dość trudne pytanie o potencjalną szkodę. Czasami mogą pojawić się wczesne objawy złożonych problemów, których nie można jeszcze w pełni zdefiniować ani ująć w ramy zasad i progów monitorowania. Próby włamań stanowią dobry przykład obejmujący ten punkt widzenia. Możesz na przykład napotkać próby phishingu skierowane przeciwko Tobie lub Twoim współpracownikom albo ktoś może paść ofiarą oprogramowania ransomware lub wirusów. Chociaż problemy te można wyodrębnić lub ograniczyć i niekoniecznie stanowią one same w sobie szerszy problem środowiskowy, w takich sytuacjach krytyczny może być czas.

Podobnie jak wiele firm, w naszej organizacji używaliśmy programu Skype dla firm firmy Microsoft jako narzędzia do komunikacji błyskawicznej i czatu. Któregoś dnia otrzymałem zaproszenie od CTO na spotkanie. Wydało mi się to niezwykłe, ponieważ CTO i ja nigdy wcześniej nie rozmawialiśmy i spodziewałem się, że jakiekolwiek spotkanie będzie koordynowane przez jego asystenta. Wiadomość z zaproszeniem była również dość niejasna. Choć nie wydarzyło się jeszcze nic szkodliwego, zdecydowałem się skontaktować mailowo z CTO i sprawdzić, czy rzeczywiście to on zainicjował ze mną kontakt. Kiedy dowiedziałem się, że tak się nie stało, powiadomiliśmy zespół IS o możliwym ataku typu

phishing i do wszystkich pracowników firmy wysłano komunikat informujący o próbach wciągania ludzi w rozmowy, podczas których mogliby ujawnić poufne informacje.

Wniosek

Szkoda jest integralną i nieuniknioną częścią życia IT. Jako taki należy go obliczyć i zintegrować z polityką i procedurami pracy, stosując zasadę projektowania uwzględniającego awarie. Każdy rodzaj szkody wymaga własnych mechanizmów zapobiegania, identyfikowania, a następnie łagodzenia, jeśli to możliwe. Jednak urządzenia i oprogramowanie mogą zdziałać tylko tyle, jeśli mają do czynienia z ludzkim uporem i pomysłowością, dlatego etyka pełni rolę spoiwa między działaniami a konsekwencjami. W trakcie codziennej pracy należy spojrzeć na każde zadanie przez pryzmat szkody i określić, czy istnieje ryzyko wystąpienia szkód lub niekontrolowanych skutków. Jeśli uważasz, że Twoja praca może wywołać negatywny wynik, powinieneś postępować etycznie i zapobiegać szkodom. Dwa kluczowe elementy tego równania obejmują określenie punktu bez powrotu i plan wycofania, który powinien pozwolić na przywrócenie normalnego działania, gdyby coś poszło nie tak. Niemniej jednak przypadkowe szkody nadal będą się zdarzać. Kiedy to nastąpi, ważne jest, aby aktywnie zarządzać szkodami – odpowiedzialność i aktualne informacje mogą wiele zdziałać, aby złagodzić szkody. Czasami przyznanie się do błędów lub poproszenie innych o pomoc może być bardzo trudne, ale w szerszym zakresie mogą one stanowić różnicę między uczciwymi błędami a poważnymi naruszeniami etyki.