Pierwszą rzeczą, o którą należy zapytać, jest to, dlaczego używasz starego oprogramowania? Czy to dlatego, że nie jesteś w stanie zaktualizować oprogramowania serwera? A może jest to spowodowane starymi zależnościami, z których korzysta twoje oprogramowanie? A może jest to spowodowane brakiem funduszy na aktualizację? Ale przyjrzymy się każdemu z tych przypadków, ponieważ każdy z nich stanowi inne zagrożenie dla biznesu i może mieć inne rozwiązanie.

W tym artykule chcę skupić się tylko na oprogramowaniu online, ponieważ kwestia głębi technicznej ma największy wpływ na oprogramowanie online. Opiszę również, jakby był to przestarzały problem PHP, ale w rzeczywistości może on dotyczyć dowolnego języka programowania i dowolnego skryptu.

Jak skończyłeś z długiem technicznym?

Istnieje wiele powodów, dla których można skończyć z długiem technicznym. Jednym z powodów może być po prostu brak pieniędzy na aktualizację, ale w rzeczywistości aktualizacje nie powinny być bardzo drogie. Jeśli więc nie możesz sobie pozwolić na regularne aktualizacje, być może Twoja firma nie zarabia wystarczająco dużo pieniędzy. Inną rzeczą może być brak procesu biznesowego do aktualizacji wszystkiego, a brak tej procedury może mieć wpływ na wiele obszarów, nie tylko ten, który jest przestarzały w danym momencie, ponieważ później mogą pojawić się inne. Trzecim scenariuszem, który mógłbym sobie wyobrazić, jest to, że otrzymałeś lub kupiłeś przestarzały skrypt, który wymaga teraz aktualizacji, więc koszt inwestycji wzrośnie o kwotę kosztu aktualizacji.

Bardzo ważne jest, aby znaleźć główną przyczynę nieaktualnego oprogramowania, ponieważ pozwoli to również uniknąć podobnych problemów w przyszłości. Jeśli więc chcesz to zrobić poprawnie, cała procedura powinna rozpocząć się od zidentyfikowania przyczyny źródłowej i jej zmiany (lub wdrożenia). W rezultacie jakość Twojej firmy ulegnie poprawie, dzięki czemu będzie mogła prosperować i dalej się rozwijać.

Jak stare oprogramowanie wpływa na użyteczność biznesową SaaS?

Istnieje wiele aspektów, w których korzystanie ze

starego oprogramowania wpływa na biznes. Pierwszym z takich aspektów jest oczywiście bezpieczeństwo. Jeśli nie korzystasz z najnowszego oprogramowania, możliwe, że nie otrzymałeś najnowszych aktualizacji zabezpieczeń. Nieotrzymywanie aktualizacji bezpieczeństwa może mieć wpływ zarówno na użytkowników, jak i na firmę i pracowników. W każdym z tych przypadków może dojść do kradzieży przetwarzanych danych. Innym skutkiem niebezpiecznego oprogramowania może być to, że po prostu przestaje działać, gdy inne zależności są aktualizowane.

Drugim aspektem jest użyteczność. Bez najnowszych aktualizacji prawdopodobnie nie otrzymujesz nowych funkcji, a także poprawek do starych. Taka nowa funkcja może potencjalnie poprawić doświadczenie użytkownika lub pracowników, a tym samym pozytywnie wpłynąć na wyniki biznesowe. Na przykład nowa funkcja może przyspieszyć procesy dla pracowników, poprawiając w ten sposób ich wydajność, a ostatecznie samą firmę.

Użyteczność zależy również od szybkości

oprogramowania. Nowe oprogramowanie często wprowadza również poprawki wydajności. Jeśli oprogramowanie jest opóźnione, użytkownicy stają się niecierpliwi, źli i sfrustrowani. W miarę jak frustracja użytkownika rośnie, może on zdecydować, że nie będzie już w ogóle korzystał z tego oprogramowania.

Oczywiście, są to tylko przykłady, ale myślę, że przekonałem cię, że aktualizacja oprogramowania jest bardzo ważna dla samego biznesu.

Czy używanie przestarzałych wersji PHP jest niebezpieczne dla mojej strony internetowej?

Najbardziej ogólną odpowiedzią na to pytanie byłoby: to zależy, ale generalnie tak.

Najgorszym przypadkiem jest –

bez względu na poziom błędu – wyświetlanie komunikatów o błędach na stronie głównej lub blokowanie części funkcjonalności witryny (to NIGDY nie powinno się zdarzyć – ponieważ NAPEWNO wyłączyłeś wyświetlanie błędów na produkcji i będzie to widoczne tylko w dziennikach! W przeciwnym razie może to być OGROMNY problem z bezpieczeństwem!). Choć nie brzmi to groźnie, są to informacje, które udostępniasz potencjalnemu atakującemu za darmo. I nie jest to rzadkie do

zaobserwowania. W skrajnym przypadku, jeśli dane wyświetlane na pierwszej stronie ujawniają zbyt wiele, haker nie musi nawet atakować – po prostu loguje się do panelu administracyjnego. Nigdy nie należy wyświetlać w przeglądarce błędów zaplecza w oprogramowaniu produkcyjnym.

Istnieje kilka poziomów błędów php, z których najczęstsze to: deprecated, notice, warning i error. W przypadku najmniejszego wpływu, podczas aktualizacji php do nowszej

wersji, zobaczysz przestarzałe błędy. Oznacza to, że w przyszłych wersjach korzystanie z takiej funkcjonalności ulegnie zmianie. Powinieneś zaktualizować jak najszybciej, ponieważ wraz z kolejną wersją twoje oprogramowanie przestanie działać – więc bez aktualizacji skryptu nie jesteś w stanie zaktualizować całego oprogramowania.

Nieco gorszym przypadkiem jest sytuacja, w której zignorowałeś komunikat o deprecjacji lub go przeoczyłeś, a teraz po

aktualizacji oprogramowanie w ogóle przestało działać. W takim przypadku najczęściej otrzymasz ostrzeżenia lub błędy i nie będziesz w stanie kontynuować korzystania z oprogramowania. W takiej sytuacji można jedynie powrócić do starej wersji PHP lub zaktualizować skrypt tak szybko, jak to możliwe, aby witryna znów działała.

Możliwość przywrócenia aktualizacji oprogramowania jest bardzo ważna w rozwoju oprogramowania. Jak widać na powyższym przykładzie, dlatego

w Sailing Byte zawsze staramy się zachować taką możliwość. Najmniej wygodnym w takim przypadku, ale wciąż akceptowalnym rozwiązaniem powinno być przywracanie z kopii zapasowych, które zawsze powinieneś mieć.

Jakiej wersji PHP powinienem używać?

Najprostszą odpowiedzią byłoby: najnowsza, na której twój skrypt jest w stanie działać. Najlepiej, oczywiście, gdyby była to

najnowsza dostępna wersja PHP, która jest w stabilnej gałęzi. Jeśli zadajesz to pytanie, to prawdopodobnie powinieneś również zapoznać się z ulepszeniami w najnowszych wersjach PHP.

Dostawcy usług hostingowych w większości przypadków nie oferują pomocy przy migracji starych skryptów do nowej wersji PHP. Nawet jeśli pozwalają na zmianę wersji PHP. Ale czy powinieneś używać starszej wersji, która nie obsługuje zabezpieczeń?

Oczywista odpowiedź brzmi – nie.

I dla przypomnienia – PHP 5.x nie jest już rozwijane i nie otrzymuje aktualizacji bezpieczeństwa. PHP 6 nigdy nie zostało wdrożone. Nawet cała gałąź PHP 7.x nie będzie już otrzymywać żadnych aktualizacji. Powinieneś również zawsze unikać używania wersji php 8.0, ponieważ nie otrzymuje ona już aktualizacji bezpieczeństwa – więc w tej chwili musisz celować powyżej 8.0. Dopuszczalne wersje PHP to 8.1 i 8.2, ale preferowane byłyby

8.3 lub 8.4. Możesz sprawdzić oficjalną stronę php.net dla aktywnego wsparcia i bezpieczeństwa. daty wsparcia, ale dla Twojej wygody załączam tabelę poniżej.

Oddział	Wstępne wydanie	Aktywne wsparcie do	Wsparcie bezpieczeństwa do
8.1	25 listopada 2021	25 listopada 2023	31 grudnia 2025
8.2	8 grudnia 2022	31 grudnia 2024	31 grudnia 2026
8.3	23 listopada 2023	31 grudnia 2025	31 grudnia 2027
8.4	21 listopada 2024	31 grudnia 2026	31 grudnia 2028

Jeśli więc nie wiesz, jak dokonać aktualizacji lub szukasz kogoś, kto zrobi to za Ciebie – kliknij poniższy formularz kontaktowy i daj nam znać. Możemy Ci pomóc!

Jak zmniejszyć wpływ aktualizacji oprogramowania?

Przede wszystkim należy zawsze mieć kopię zapasową i możliwość przywrócenia. Nie rozpoczynaj nawet żadnych prób, zanim nie upewnisz się, że jesteś w stanie przywrócić zmiany.

Następne pytanie brzmi: jaki jest akceptowalny czas przestoju dla aktualizacji? Jeśli prowadzisz małą stronę internetową,

prawdopodobnie możesz pozwolić sobie na bycie offline nawet przez kilka godzin. Ale jeśli prowadzisz dużą usługę SaaS, możesz rozważyć aktualizację w godzinach bezczynności (np. w nocy) lub stworzyć niebiesko-zielony system aktualizacji z płynnym przełączaniem za odwrotnym proxy z mechanizmem awaryjnym. W przypadku większości usług w fazie rozwoju prawdopodobnie wystarczające będzie posiadanie odpowiedniego zautomatyzowanego potoku

CI/CD z wbudowanym skryptem aktualizacji o minimalnym czasie przestoju. Jeśli nie jesteś pewien, co tutaj opisuję i chcesz to zrobić samodzielnie – po prostu wybierz pierwszą opcję.

Gdy planowany jest przestój, należy również poinformować o tym użytkowników. Możesz rozważyć wysłanie do nich wiadomości e-mail lub przynajmniej pozostawienie strony „konserwacja” na stronie głównej – aby mogli być pewni, że wszystko jest w porządku, a aktualizacja jest planowana.

Następną rzeczą, którą powinieneś rozważyć, jest oczywiście najpierw przetestowanie. Zanim „oszalejesz” na produkcji, najpierw spróbuj przejść przez cały proces na stronie testowej (stwórz jedną dla inscenizacji, jeśli w tej chwili jej nie masz!). Podczas aktualizacji do nowej wersji PHP zdecydowanie musisz najpierw sprawdzić swoją witrynę na oddzielnym serwerze testowym. Następnie możesz ocenić ilość pracy potrzebnej do wprowadzenia pełnej funkcjonalności na stronie

produkcyjnej – jeśli w ogóle jest to konieczne. Czasami, jeśli oprogramowanie zostało odpowiednio zaktualizowane, witryna nie będzie miała problemów z przejściem na nową wersję PHP. Na koniec, przetestuj wszystko. Jeśli wszystko jest w porządku – przejdź dalej.

Teraz w oparciu o twoje doświadczenia podczas procesu aktualizacji na stronie testowej, ponieważ prawdopodobnie napotkałeś tam pewne problemy, upewnij się, że masz

przygotowaną kompletną i odpowiednią listę kontrolną z notatkami na temat rozwiązywania potencjalnych problemów. Jeśli są jakieś błędy lub użycie przestarzałych funkcji, należy je dokładnie sprawdzić. Ważne jest, aby naprawić wszystkie „przestarzałe” powiadomienia. Jest to informacja dla Ciebie, że w następnych wersjach PHP funkcje nie będą obecne – i strona po prostu przestanie działać. Twoje notatki mogą ci bardzo pomóc podczas aktualizacji na produkcji – uwierz

mi, wiem co mówię!

Jeśli podczas procesu aktualizacji będziesz musiał również zaktualizować ustawienia domeny, upewnij się, że przed planowaną aktualizacją co najmniej tydzień wcześniej zmniejszysz ustawienia TTL. Zapewni to, że podczas zmian w ustawieniach rekordów DNS będą one propagowane z najlepszą możliwą prędkością na całym świecie.

Oczywiście, po aktualizacji. nawet jeśli uważasz, że wszystko

poszło pomyślnie, powinieneś sam wszystko przetestować. Cały proces, przez który przechodzą użytkownicy, zadania cron, wysyłanie wiadomości e-mail, płatności itp. Powinieneś także sprawdzić, czy twoje oprogramowanie do monitorowania zgłoszeń działa dobrze i akceptuje nowe zgłoszenia, więc gdy użytkownik uruchomi coś, co nie zostało uwzględnione w testach lub pominięte, natychmiast się o tym dowiesz. Jest to jeden z podstawowych elementów oprogramowania, z którego

regularnie korzystamy w Sailing Byte.

Jeśli obejmiesz wszystkie powyższe punkty, powinieneś być dobry z aktualizacją.

Porządek aktualizacji dla WordPress i PHP

Pamiętaj, że zawsze możesz znaleźć informacje o obsługiwanych wersjach php na oficjalnej stronie wordpress.org. Ale dla wygody odtworzyłem poniższą tabelę kompatybilności.

SAILINGBYTE

WordPress	PHP 8.4	PHP 8.3	PHP 8.2	PHP 8.1	PHP 8.0	PHP 7.4	PHP 7.3	PHP 7.2	PHP 7.1	PHP 7.0	PHP 5.6
6.7	Tak, Beta	Tak, Beta	Tak, wyjątki	Tak, wyjątki	Tak, wyjątki	Tak	Tak	Tak	Nie	Nie	Nie
6.6	Nie	Tak*	Tak, wyjątki	Tak, wyjątki	Tak, wyjątki	Tak	Tak	Tak	Nie	Nie	Nie
6.5	Nie	Tak, Beta	Tak, Beta	Tak, wyjątki	Tak, wyjątki	Tak	Tak	Tak	Tak	Tak	Nie
6.4	Nie	Tak, Beta	Tak, Beta	Tak, wyjątki	Tak, wyjątki	Tak	Tak	Tak	Tak	Tak	Nie
6.3	Nie	Nie	Tak, Beta	Tak, wyjątki	Tak, wyjątki	Tak	Tak	Tak	Tak	Tak	Nie
6.2	Nie	Nie	Tak, Beta	Tak, Beta	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
6.1	Nie	Nie	Tak, Beta	Tak, Beta	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
6.0	Nie	Nie	Nie	Tak, Beta	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
5.9	Nie	Nie	Nie	Tak, Beta	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
5.8	Nie	Nie	Nie	Nie	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
5.7	Nie	Nie	Nie	Nie	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
5.6	Nie	Nie	Nie	Nie	Tak, Beta	Tak	Tak	Tak	Tak	Tak	Tak
5,5	Nie	Nie	Nie	Nie	Nie	Tak	Tak	Tak	Tak	Tak	Tak
5.4	Nie	Nie	Nie	Nie	Nie	Tak	Tak	Tak	Tak	Tak	Tak
5.3	Nie	Nie	Nie	Nie	Nie	Tak	Tak	Tak	Tak	Tak	Tak

Ta tabela powinna być pomocna w tworzeniu planowanego procesu aktualizacji, nawet jeśli

używasz bardzo starych wersji PHP i WordPress. Przykładowa ścieżka może wyglądać następująco, jeśli używasz wersji WordPress. 5.5 i php w wersji 7.4:

1. **zaktualizuj WordPress, na przykład do wersji 6.0**
2. **Zwiększ wersję PHP do 8.0 lub 8.1**
3. **Zaktualizuj WordPress do wersji 6.6 lub nawet 6.7**
4. **Postaraj się używać PHP 8.3 lub 8.4**

SAILINGBYTE

Zaktualizuj Laravel i jego kompatybilność z PHP

Ponownie, dla Twojej wygody, odtworzyłem tabelę opartą na dokumentacji Laravel, abyś mógł sam zobaczyć, jakie wersje PHP są obsługiwane przez które wersje Laravel.

Wersja Laravel	PHP 5.6	PHP 7.0	PHP 7.1	PHP 7.2	PHP 7.3	PHP 7.4	PHP 8.0	PHP 8.1	PHP 8.2	PHP 8.3	PHP 8.4
Laravel 5.0	Tak	Tak									
Laravel 5.1	Tak	Tak									
Laravel 5.2	Tak	Tak									
Laravel 5.3	Tak	Tak									
Laravel 5.4	Tak	Tak	Tak								
Laravel 5.5		Tak	Tak								
Laravel 5.6			Tak	Tak							
Laravel 5.7			Tak	Tak							

Laravel 5.8			Tak	Tak	Tak						
Laravel 6.x				Tak	Tak	Tak					
Laravel 7.x				Tak	Tak	Tak					
Laravel 8.x					Tak	Tak	Tak				
Laravel 9.x							Tak	Tak	Tak		
Laravel 10.x								Tak	Tak	Tak	
Laravel 11.x									Tak	Tak	Tak
Laravel 12.x									Tak	Tak	Tak

Powinieneś czytać tę tabelę w podobny sposób, w jaki czytaliśmy tabelę aktualizacji WordPress. Zasadniczo aktualizuj na przemian Laraval i PHP. Upewnij się tylko, że używasz kompatybilnych wersji. Oczywiście podczas aktualizacji Laravel do następnej wersji należy zawsze postępować zgodnie z oficjalną listą kontrolną. W podobny sposób

należy postępować zgodnie z listą kontrolną aktualizacji PHP między każdą wersją.

Aktualizacja – czy warto?

Nie powinno być zaskoczeniem, jeśli powiem, że aktualizacja jest nie tylko warta, ale i konieczna. Ale jeśli zrozumiałeś ten artykuł, nie powinno to być dla ciebie zaskoczeniem.

Pamiętaj jednak, że opisałem tutaj wszystko tylko z

perspektywy PHP, ale zależności może być znacznie więcej. Dla WordPressa mogą to być wtyczki. Dla front-endu mogą to być na przykład pakiety React JS lub biblioteki JQuery, a dla bazy danych mogą to być wersje MySQL lub MariaDB. Nie wspominając o tym, że powinieneś również zadbać o aktualizację pakietów repozytorium dystrybucji linux – więc jest całkiem sporo do omówienia.

Oczywiście, może to być trochę za dużo informacji do

przetworzenia dla osoby nietechnicznej, więc jest to dobry czas, aby zapytać osobę, która opiekuje się twoją witryną, czy możesz zaktualizować wersję PHP – najlepiej do najnowszej. Upewnij się również, że wdrożyłeś już proces aktualizacji w swoich procedurach biznesowych. Jeśli jednak nie masz takiej osoby – nie wahaj się skontaktować z nami za pośrednictwem poniższego formularza kontaktowego i zacznijmy współpracę!