Kodowanie niestandardowe czy nadpisywanie wtyczek?

Przygotowanie strony internetowej tak, aby była jak najwygodniejsza w użyciu, jest jednym z celów niestandardowego kodowania. W ten sposób możesz zapewnić, że doświadczenie użytkownika jest w 100% takie, jak zostało zaprojektowane w Twojej strategii. Pozwala także być **lepszym** niż konkurencja w dostarczaniu dokładnie tego, czego potrzebuje klient. Czasami używasz wtyczki Joomla lub WordPress, która jest **prawie idealna** – nadpisanie rdzenia pozwoliłoby ci zapewnić, że jest ona w pełni doskonała. Czy powinieneś to zrobić?

Jude była młodą programistką – uwielbiała tworzyć strony internetowe na WordPressie. Była niedoświadczona, ale obiecująca i miała coraz więcej klientów. Chcieli, aby nie tylko kodowała ich strony internetowe, ale także opiekowała się nimi. We wtorek otrzymała telefon: *Hej, czy chciałabyś zająć się moją stroną internetową? Wymaga ona kilku aktualizacji i drobnych poprawek.* Oczywiście, zgodziła się i wkrótce potem otrzymała dane do logowania. Witryna nie była naprawdę zadbana w przeszłości – wiele wtyczek i rdzeni było przestarzałych, co czyniło ją podatną na ataki hakerów. Jude zgodził się na stawkę i zaczął aktualizować rdzeń i wtyczki.

Pod koniec dnia wszystko było gotowe – ale otrzymała **telefon**. *Co się dzieje ze stroną internetową? Formularz kontaktowy wygląda zupełnie inaczej! Również posty biznesowe nie wyświetlają połowy informacji!*

Co do diabła – pomyślała Jude. Ponieważ była niedoświadczona, nigdy nie spotkała się z tak złą praktyką, jak hakowanie podstawowych plików. Niestety, to właśnie pułapka, którą zostawił dla niej poprzedni deweloper. **wstronie nie dało się zaktualizować** bez ręcznego wyszukania plików, sprawdzenia różnic i zaaplikowania ich do najnowszych wersji wtyczek! Co więcej, ani wtyczki nie zostały napisane poprawnie, ani hacki nie zostały wykonane zgodnie z modelem frameworka. Sytuacja ta doprowadziła do **sporego** zamieszania.

Na szczęście miała już odrobioną pracę domową. Zmęczona i wściekła po godzinach pracy, była w stanie przywrócić kopię zapasową i wrócić do domu z obietnicą dla siebie – jeśli kiedykolwiek dowie się, kto to zrobił, pożałuje tego.

Jak widać w tym przypadku, Jude została postawiona w dość trudnej sytuacji. Z jednej strony, miała stronę internetową, która wymagała aktualizacji, z drugiej strony, wiele zostało zhakowanych w rdzeniu przez poprzedniego dewelopera. Poprzedni programista był bezmyślnym facetem, który chciał wykonać swoją pracę szybko i tanio. Prowadzi to do wniosku, że hakowanie rdzenia doprowadzi go do celu, ponieważ właściwe podejście zajęłoby dwa razy więcej czasu i pieniędzy. Osiągnął swój cel **ale pozostawił witrynę niemożliwą do utrzymania i podatną na ataki**.

Czy można dokonać pewnych zmian w kodzie, gdy nie jest się właścicielem wtyczki?

Czy istnieją inne sposoby poza hakowaniem rdzenia, aby dokonać prostych zmian w kodzie, jeśli nie jesteś właścicielem wtyczki? Oczywiście, że są:

- **Możesz współpracować z oryginalnym deweloperem. Wyślij mu wiadomość z przygotowanymi zmianami w rdzeniu. Jeśli deweloper jest opiekuńczy, przejrzy twój kod i dołączy go do głównej wtyczki. W następnej wersji wtyczki będziesz mieć funkcjonalność, której potrzebujesz! Minusem jest to, że minie trochę czasu, zanim zmiany pojawią się w publicznym repozytorium**

  .

- **Rozszerzenie za pomocą osobnej wtyczki – możesz pozostawić oryginalną wtyczkę nietkniętą i napisać własną, która będzie działać równolegle z oryginalną. Takie podejście nie zawsze jest możliwe, jeśli potrzebne są zmiany wewnątrz rdzenia**
- **Użyj niestandardowych haków – ta metoda jest przeznaczona tylko dla WordPress. Haki pozwalają dostosować zachowanie wtyczki. Ale – bądź świadomy – ta metoda może być używana tylko wtedy, gdy oryginalny programista przygotował haki! I niestety, nie jest to tak powszechne, aby wtyczki WordPress miały odpowiednie haki**

  .

- **Nadpisywanie wywołań zwrotnych – metoda dla WordPress. Chociaż na pierwszy rzut oka może się to wydawać dobrym pomysłem, ma również wadę. Często funkcje nie są wystarczająco modularne, aby wywoływać tylko te fragmenty kodu, na których Ci zależy**

  .

- **Dodawanie niestandardowych haków – metoda dla WordPress. Jest to rozwiązanie pomiędzy nadpisywaniem wywołań zwrotnych a współpracą z deweloperem. W tym przypadku nadpisujesz podstawowe pliki swoimi hakami, a następnie wywołujesz je z osobnych plików. Przy aktualizacji nadal musisz wykonać niestandardową pracę, chociaż jest to tylko dodanie własnych haków z powrotem – chyba że oryginalny programista doda twoją poprawkę do strony głównej**

  .

- **Fork! – dla każdego CMS. Jeśli licencja wtyczki na to pozwala, możesz skopiować kod i stworzyć własną wersję wtyczki. Ale stracisz możliwość aktualizacji wtyczki, jeśli jakiekolwiek aktualizacje zostaną opublikowane przez oryginalnego autora!**
- **MVC overrides – metoda dla Joomla. Jest to bardzo potężna metoda, ponieważ pozwala na nadpisywanie widoków lub całych plików bez dotykania rdzenia. Jest również bardzo wygodna w użyciu i nie wymaga żadnego rodzaju haków ani wywołań zwrotnych.**

Podsumowanie

Ogólnie rzecz biorąc, podejście, które powinieneś wybrać, zależy od wielu czynników. Czy włamanie jest tylko tymczasowe? Czy przechodzisz na inny CMS w krótkim czasie (możesz przeczytać o zaletach i wadach różnych CMS tutaj)? Czy chcesz zachować kompatybilność z aktualizacjami wtyczek lub rdzenia w dłuższej perspektywie? Czy wtyczki powinny być utrzymywane ze względów bezpieczeństwa, czy może powinny być używane na innych stronach internetowych? Czy chcesz wymienić nowe funkcje i bezpieczeństwo na szybki hack? Odpowiedzi na te pytania pozwolą ci dokonać właściwego wyboru.