

Czym jest dług techniczny?

Podczas całego procesu tworzenia oprogramowania należy podjąć wiele decyzji zarówno technologicznych, jak i organizacyjnych/biznesowych. Właściwy proces decyzyjny powinien poprzedzać faktyczną pracę & wykonanie, aby zapobiec występowaniu przyszłych problemów i możliwych błędów.

>

Działa to **tylko w teorii**. W prawdziwym życiu rzadko mamy luksus poświęcenia odpowiedniej ilości zasobów (w tym czasu) na ten proces. Ta presja popycha nas do godzenia się na nieuniknione kompromisy i ostatecznie do realizacji wielu nieoptymalnych rozwiązań (lub czasami do nierealizowania żadnych rozwiązań/zmian, gdy są one wyraźnie potrzebne).

Dług techniczny to stale rosnąca suma “kosztów” wszystkich tych nieuniknionych, nieoptymalnych decyzji (lub ich braku).

Intencjonalne / nieintencjonalne przyczyny długu technicznego

Powiedzmy, że musimy zdecydować między rozwojem pewnego produktu przy użyciu starej, taniej i szybkiej do zakodowania technologii/rozwiązania “A” lub nowo wynalezionej, ale dość drogiej i wymagającej czasochłonnego kodowania technologii/rozwiązania “B”.

Wybierając A wygenerujemy pewne oszczędności w krótkim okresie, ale ostatecznie będziemy musieli pokryć ogromne koszty albo ciągłego debugowania (co spowolni nowy rozwój), albo całkowicie przepisać stare oprogramowanie przy użyciu nowoczesnej technologii (co potencjalnie zatrzyma nowy rozwój z powodu migracji).

Wybór technologii/rozwiązania B daje nam długoterminową stabilność i możliwości rozwoju, ale płacimy za to wyższą ceną rozwoju oprogramowania.

Wybór A lub B oznacza, że **intencjonalnie** akceptujemy fakt, że nasze oprogramowanie będzie gromadzić Dług Techniczny szybciej lub wolniej. W tym przypadku jesteśmy przynajmniej w stanie proaktywnie zająć się nadchodzącymi problemami.

Z drugiej strony, brak dbałości o podejmowanie decyzji i słaba jakość wykonania skutkuje powstaniem **niezamierzonego** długu technicznego, który pojawi się niezależnie od wyboru A lub B. Zatrudnianie profesjonalistów i utrzymywanie wysokich standardów komunikacji z nimi pomoże ci uniknąć tego w jak największym stopniu.

	Intentional cause	Unintentional cause
Technical reasons	When code or design are too complex, so to avoid breakdown or missing a deadline a conscious decision for a quick fix is made.	Poor design or low-quality code delivered due to incompetence, lack of standards, guidelines or team coordination
Organizational reasons	Business logic, contracts or deadlines to release a product or a feature dictate an intentional decision to go into technical debt.	Bad management, work ethics, software development processes, or other organizational issues lead to technical debt in terms of productivity.

Jak radzić sobie z długiem technicznym?

1) Zdefiniuj go i śledź poprzez wizualizację w Product Backlog.

“Jeśli czegoś nie ma w Backlogu Produktu, to nie istnieje”

Zasada ta powinna mieć zastosowanie nie tylko

do przyszłych planów rozwoju i małych błędów, ale także do szerszych kwestii długu technicznego, takich jak na przykład:

“przepisanie / refaktoryzacja kodu funkcji X/Y/Z w celu zapewnienia zgodności z najnowszymi zmianami w architekturze produktu w celu utrzymania najwyższej wydajności aplikacji podczas szybkiego skalowania liczby użytkowników końcowych”

2) Stworzenie planu i jego wykonanie.

>

Po zwizualizowaniu Długu Technicznego w Backlogu Produktu możesz skupić się na tym, co jest krytyczne do “spłacenia”. Nie spiesz się i omów to zarówno z perspektywy biznesowej (ryzyko, wartość dla klienta), jak i technicznej (kwestie wydajności kodu). Po zdefiniowaniu, gdzie pomoc jest najbardziej potrzebna, masz dwa rozwiązania:

a) podziel słonia na sprinty (metoda ewolucyjna)

W większości przypadków zawsze istnieje możliwość zaplanowania sprintu z uwzględnieniem zarówno nowego rozwoju, jak i zadań związanych z obsługą długu technicznego. Tak - oznacza to wolniejszy rozwój, ale oznacza to również bardziej stabilny, przewidywalny, bezpieczny i skalowalny rozwój!

b) przygotowanie wyłącznego sprintu konserwacyjnego i/lub migracyjnego (metoda rewolucyjna)

W przypadkach, w których nie ma sensu pracować krok po kroku (na przykład ogromne ryzyko wystąpienia krytycznych problemów z wydajnością produktu), należy rozważyć przeniesienie całej uwagi na Dług Techniczny i zaprzestanie jakiegokolwiek “nowego” rozwoju.

3) Zapobieganie gromadzeniu się Długu Technicznego

>

**Im bardziej skupiasz się na jakości:
- procesów decyzyjnych (ocena ryzyka itp.);**

- komunikacji z deweloperami Twojego produktu (omawianie wewnętrznych procedur, bycie otwartym na szczerą informację zwrotną na temat pewnych rozwiązań itp.) ;

tym mniej Długu Technicznego będzie się gromadzić.

Jeśli uważasz ten artykuł za pomocny, rozważ poszukanie ich więcej w naszej Bazie Wiedzy Sailing Byte's!

>