Mówimy dużo o rozwoju aplikacji i stron internetowych, a także o programistach. Programista front-end zajmuje się interfejsem, a programista back-end dba o funkcjonalność. Wiemy, jakich narzędzi używają i jakie są składniki oprogramowania. Musimy jednak pamiętać, że „na początku było słowo”. W IT słowo to oznacza język kodowania. Dlatego też w dzisiejszym artykule chciałbym skupić się na głównej czynności jaką wykonują programiści – kodowaniu.

Co to jest kodowanie?

.

Aby właściwie odpowiedzieć na pytanie, czym jest kodowanie, wyobraźmy sobie rzeczywistą sytuację. Dwie osoby mówiące różnymi językami i chcące się komunikować mają dwie opcje. Mogą albo nauczyć się mówić w trzecim wspólnym języku, albo jedna z nich nauczy się mówić w języku drugiej osoby. Wyobraźmy sobie teraz taką samą sytuację między człowiekiem a komputerem. Aby skutecznie komunikować się z komputerem, ludzie musieli stworzyć specjalny język. Kodowanie to użycie tego specjalnego języka do napisania zestawu instrukcji. Instrukcje te skutkują stworzeniem oprogramowania, strony internetowej lub aplikacji.

Kodowanie frontend vs backend – jaka jest różnica?

Zrozumiałe jest, że kodowanie w programowaniu front-end będzie się różnić od kodowania w programowaniu back-end. Programista front-end używa języków programowania, które poprawiają wizualną część strony internetowej. Programista back-end koduje w językach programowania, które zapewniają funkcjonalność danego oprogramowania. Języki te nie są jednak takie same. Niektóre elementy lub polecenia mogą być podobne, ale konstrukcja różni się tak samo, jak użycie.

Przykłady języków front-end:

- HyperText Markup Language (HTML)
- Cascading Style Sheets (CSS)
- JavaScript
- React
- Angular
- Vue

- jQuery
- Swift

Przykłady języków back-end:

- C#
- Golang (Go)
- Java
- JavaScript
- PHP
- Python
- Ruby
- SQL

Co to są metodologie kodowania?

Podobnie jak w wielu innych dziedzinach życia, istnieją różne sposoby podejścia do kodowania. Takie podejścia nazywane są metodologiami programowania. Na przestrzeni lat powstało wiele różnych metodologii kodowania. Wszystkie mają wspólny cel, jakim jest znalezienie najlepszego sposobu na poprawę jakości kodu, produktywności zespołu i współpracy. Przyjrzyjmy się niektórym z najważniejszych metodologii w tworzeniu oprogramowania.

>

Rozwój wodospadu

Model kaskadowy jest podstawową i najstarszą metodologią rozwoju produktu. Opiera się na przekonaniu, że rozwój projektu składa się z sekwencyjnych faz. Kolejna rozpoczyna się po zakończeniu poprzedniej. Jest uważany za jedną z mniej elastycznych metodologii i przeciwieństwo zwinnego rozwoju.

>

Agile development

Agile jest jedną z najbardziej znanych metodyk rozwoju oprogramowania XXI wieku. Została stworzona w 2001 roku przez kilku programistów podpisujących Manifest Agile. Manifest składa się z dwunastu zasad dotyczących prowadzenia rozwoju.

Podstawowe zasady opierają się na prostocie i ciągłym rozwoju produktu poprzez współpracę i ewolucję wymagań. W ramach metodologii Agile istnieje kilka metod koncentrujących się na różnych czynnikach, takich jak praktyki czy przepływ pracy.

>

Metoda dynamicznego rozwoju systemów (DSDM)

DSDM kładzie nacisk na poprawę współpracy, a nie technologii. Według tej metody, więcej projektów kończy się niepowodzeniem z powodu błędów ludzi we współpracy.

Kanban

Bardzo popularna metoda wizualizacji zadań, wydajności pracy i postępów. Wiele osób może kojarzyć nazwę Kanban z tablicami Kanban. Są one uproszczoną wersją idei stojącej za rozwojem Kanban. Tablice Kanban sprawdzają się wszędzie tam, gdzie potrzebne jest usprawnienie zarządzania projektami.

Scrum

Ramuła Scrum jest powszechnie stosowana nie tylko w tworzeniu oprogramowania, ale także w sprzedaży lub marketingu. Scrum opiera się na sprintach, które są ramami czasowymi, w których zespół osiąga małe zadania lub cele. Takie sprinty trwają od 2 do 4 tygodni.

Oszczędne tworzenie oprogramowania

Metodyka lean opiera się na przekonaniu, że wszystko, co nie stanowi wartości dla klienta, jest marnotrawstwem, które należy wyeliminować. Metoda ta kieruje się także kilkoma innymi zasadami:

- Upraszczanie uczenia się
- Odwlekanie zobowiązania
- Dostarczenie tak szybko, jak to możliwe
- Wzmocnienie zespołu
- Budowanie integralności w
- Optymalizacja całości

Ciągła integracja

W tej praktyce programiści często łączą zmiany kodu z centralnym repozytorium i uruchamiają testy.

Rozwój przyrostowy

Ta metoda pozwala na budowanie systemów w przyrostach. Końcowe wymagania są znane od początku projektu.

Szybkie tworzenie aplikacji

Podejście to kładzie nacisk na adaptację, a nie planowanie projektu. Metoda ta często wykorzystuje wiele prototypów w procesie rozwoju oprogramowania.

Rozwój spiralny

Model spiralny łączy podejście kaskadowe i iteracyjne. Polega on na powtarzaniu ustalonych faz projektu i dodawaniu funkcjonalności, aż projekt będzie gotowy do wydania. Rozwój spiralny działa wyjątkowo dobrze w przypadku dużych projektów.

Shape Up

Metoda ta polega na terminowym dostarczaniu predefiniowanych projektów w ramach rozwoju produktu. Jest uważana za skuteczną metodę zarządzania ryzykiem i niepewnością.

Zaawansowane metodologie

Istnieje wiele wysokopoziomowych podejść, które koncentrują się na różnych czynnikach rozwoju oprogramowania. Rzućmy okiem na kilka najważniejszych z nich:

Model Chaosu

Metoda ta priorytetyzuje rozwiązywanie kluczowych zadań i problemów.

Powolne programowanie

Koncentruje się na precyzyjnym i stopniowym wykonywaniu zadań bez podkreślania presji czasu.

Extreme programming (XP)

>

Metodologia ta koncentruje się na poprawie jakości oprogramowania i szybkości reagowania zgodnie ze zmieniającymi się wymaganiami klienta. Charakteryzuje się częstymi wydaniami i krótkimi cyklami rozwoju.

>

Co jeśli programista popełni błąd?

Prawdopodobnie będziesz zaskoczony rewelacją, którą za chwilę ci przedstawię, ale wszyscy deweloperzy popełniają błędy. Wszyscy jesteśmy ludźmi. Kodowanie i inne procesy IT są bardzo złożone. Bez względu na to, czy jesteś programistą front-end czy back-end, nie jesteś nieomylny. Początkujący popełnia błędy, ale ekspert również. Błędy mają po prostu inny charakter ze względu na różne poziomy zaawansowania. Jak to się dzieje, że nasze oprogramowanie wciąż działa? Ponieważ wdrażamy procesy, które pomagają nam rozpoznać błędny kod i go naprawić. Część tego procesu nazywana jest przeglądem kodu.

Jak przegląd kodu zapobiega powstawaniu błędów?

Przegląd kodu to proces sprawdzania kodu źródłowego stworzonego przez programistę. Jest to czasami nazywane przeglądem partnerskim, ponieważ kod jest sprawdzany nie przez twórcę, ale przez inne osoby. Głównym celem tego działania jest identyfikacja możliwych błędów w kodowaniu. Pozwala nam to również ulepszyć kod, znaleźć alternatywne rozwiązania i służy jako narzędzie do nauki. Istnieje kilka sposobów sprawdzania kodu innych programistów:

- **pair programming and over-the-shoulder reviews** – przeglądy kodowania i współpracy w czasie rzeczywistym.
- **recenzje wspomagane narzędziami** – wykonywane za pomocą dedykowanych narzędzi, które sprawdzają dokładność kodu. Jest najbardziej skuteczny w połączeniu z programowaniem w parach lub przeglądami przez ramię.

- **email code review** – kod jest wysyłany e-mailem i sprawdzany przez recenzenta w dogodnym dla niego czasie.

Jak przegląd kodu pomaga w stabilności biznesowej?

Wyłapywanie ewentualnych błędów zapewnia funkcjonalność systemu. Przegląd kodu ma jednak jeszcze jedną zaletę, która jest bardzo ważna dla Twojej firmy. Jest nią stabilność. Wiedza o błędach z wyprzedzeniem pomaga uniknąć efektu kuli śnieżnej. Oznacza to poruszanie się tak, jakby nic się nie stało i pisanie kodu na podstawie błędu. Jak można sobie wyobrazić, im później wyłapiesz błąd, tym więcej czasu (i pieniędzy!) będziesz musiał poświęcić na jego naprawienie. Wyłapywanie błędów tak wcześnie, jak to możliwe, gwarantuje, że nie przekroczysz przewidywanych ram czasowych i budżetowych projektu. A to zapewnia stabilność biznesową. Chcesz dowiedzieć się więcej o tym, jak korzystanie z narzędzi do przeglądu kodu pomaga Twojej firmie? Sprawdź nasz post o Sentry i jego zaletach.

Błędy w kodowaniu i bugi w Twoim produkcie? Nigdy z Sailing Byte!

.

Mam nadzieję, że udało mi się wystarczająco wyjaśnić, czym jest kodowanie i jak działa przegląd kodu. W Sailing Byte przegląd kodu jest standardową praktyką. Zapewnia, że wszystkie nasze aplikacje i strony internetowe są w pełni funkcjonalne i wolne od błędów przed wydaniem. Zarezerwuj telefon, aby uzyskać wgląd w metodologię kodowania i sposób, w jaki eliminujemy błędy podczas opracowywania produktu idealnie dostosowanego do Twoich potrzeb.