

NAJLEPSZE STRATEGIE SKUTECZNYCH PROGRAMISTÓW. TRENING MISTRZOWSKI

KOD: NS

Profil uczestnika

Programista:

- posiada doświadczenie w tworzeniu systemów informatycznych;
- chce świadomie i efektywnie planować wykonywanie zadań;
- chce świadomie i efektywnie inwestować swój czas;
- chce optymalnie wykorzystywać swoje umiejętności.

Korzyści ze szkolenia

1. **Programiści optymalnie wykorzystują czas poświęcony na pracę** – ponieważ używają metod planowania zadań, które dostosowaliśmy do specyfiki pracy programistów.
2. **Czas poświęcony na rozwiązywanie problemów ulega skróceniu** – programiści, dzięki zwiększonej samoświadomości, potrafią wykształcać u siebie nowe nawyki i sposoby postępowania poprawiające efektywność pracy.
3. **Programiści mają większą motywację do pracy** – gdyż używanie technik daje poczucie kontroli nad przebiegiem realizacji zadań, a w konsekwencji większe zadowolenie w wykonywanej pracy.
4. **Programiści skutecznie komunikują się innymi członkami zespołu** – gdyż używają narzędzi efektywnej komunikacji oraz potrafią dostosować przekazywane informacje do konkretnych rozmówców.
5. **Zmniejsza się koszt realizacji projektów** – dzięki zwiększeniu efektywności pracy programistów.
6. **Zmniejsza się koszt utrzymywania i rozwijania aplikacji** – ponieważ programiści tworzą oprogramowanie wysokiej jakości.

Parametry szkolenia

CZAS TRWANIA: 3 dni – 24 godziny.

FORMA ZAJĘĆ: Laboratorium Efektywności Programisty - 50%, wykład – 50%.

WIELKOŚĆ GRUPY: do 10 osób.

Szczegółowy program

Moduły szkoleniowe	Nabyte wiedza i umiejętności, poruszane zagadnienia
Elementy komunikacji dla programistów	<ul style="list-style-type: none">• Rola komunikacji w pracy programisty• Konkretyzowanie oczekiwań – poszerzanie przestrzeni analizy• Co jest dla Ciebie ważne - docieranie do potrzeb drugiej strony• Druga pozycja percepcyjna• Ekran użytkownika jako platforma komunikacyjna• Metaprogramy<ul style="list-style-type: none">○ Ogół-szczegół○ Problem-cel○ Podobieństwa-różnice○ Autorytet wewnętrzny-zewnętrzny• Umiejętność słuchania• Wzorce i antywzorce wśród programistów – zachowania zwiększające lub zmniejszające efektywność pracy. Przykłady:<ul style="list-style-type: none">○ Problem jest na zewnątrz○ Wymyślanie koła○ Bezpieczeństwo zatrudnienia○ Reguła małych kroków○ Ograniczanie przestrzeni analizy○ Najprostsze możliwe środowisko

Planowanie zadań i organizacja pracy własnej

- Planowanie własnych zadań
 - Nawyk planowania
 - Style percepcji czasu
 - Dekompozycja zadań
 - Szacowanie zadań
 - Porównanie różnych technik szacowania
 - Techniki planowania dla programistów
 - After action review
- Zarządzanie czasem dla programistów
 - Najpierw rzeczy najważniejsze Analiza dnia pracy
 - Priorytety wewnątrz zadania
 - Microdeadlines
 - Tworzenie własnego kalendarza

Implementacja

- Analiza we/wy
- Dekompozycja algorytmu i struktury
- Równowaga ogół-szczegół
- Równowaga konkret-abstrakcja