

Program szkolenia:

Ludzie i interakcje w wysokowydajnych zespołach

Informacje:

Nazwa:	Ludzie i interakcje w wysokowydajnych zespołach
Kod:	Soft-people
Kategoria:	Agile i umiejętności miękkie Scrum Masters management architekci developerzy
Grupa docelowa:	analitycy Product Owners DevOps testerzy hr admini
Czas trwania:	1 dzień
Forma:	60% wykłady / 40% warsztaty

Jesteś managerem, team leaderem (formalnym bądź nieformalnym) pracującym z zespołem programistów. Twoja firma przeszła transformację na zwinne wytwarzanie oprogramowania, masz Scruma w jednym palcu, zespół pisze czysty kod, a w firmie wszyscy powtarzają, że autonomia, mistrzostwo i cel to klucz do motywacji pracowników.

To szkolenie idzie krok dalej, po to aby wspólnie zastanowić się co oznacza autonomia, mistrzostwo i cel na głębszym poziomie, w kontekście pracy programistów oraz czego w tym modelu brakuje. Bazując na case studies, badaniach naukowych, inspiracjach z branży i spoza niej, analizujemy czym w praktyce są "ludzie i interakcje" w wysokowydajnych zespołach.

W czasie szkolenia nauczysz się:

- Traktować przyjmowanie nowych kolegów i koleżanek do pracy jako pełnowartościową pracę pierwszej kategorii, a nie przykry obowiązek.
- Zapewniać swoim współpracownikom poczucie bezpieczeństwa w codziennej pracy i częściej zauważać gdy go brakuje.
- Wspierać budowanie umiejętności eksperckich wśród swoich współpracowników i rozpoznawać co może blokować szybsze postępy.
- Rozpoznawać dlaczego niektórym osobom trudno razem pracować i jak dostosować swój przekaz do różnych poziomów zaawansowania.
- Minimalizować wpływ błędów poznawczych przy trudnych wyborach i podejmować odrobinę lepsze decyzje technologiczne czy rekrutacyjne.
- Podnosić odpowiedzialność i zaangażowanie zespołu w realizację celów bez komenderowania i kontroli.
- Projektować środowisko pracy pod kątem responsywności a nie optymalizacji kosztowej.
- Kwestionować mainstreamowe praktyki w swoim kontekście.

Zalety szkolenia:

- Bazujemy wyłącznie na metodach potwierdzonych badaniami zgodnymi z metodą naukową

BO·TT·EGA

IT minds

- Nie bazujemy na dogmatach
- Skupiamy się na kontekście projektu informatycznego i pracy developerów, bez analogii typowych dla miękkich szkoleń

Szczegółowy program:

1. Agile development: coś poszło nie tak

2. Bezpieczeństwo

- 2.1. nawyk kluczowy
- 2.2. przykłady naruszenia bezpieczeństwa pracownika IT
- 2.3. teatr bezpieczeństwa
- 2.4. kultura blameless
- 2.5. doświadczenie pierwszego dnia
- 2.6. chartering
- 2.7. "psychological safety" i bezpieczne spotkania
- 2.8. charyzma uwagi

3. Nauka

- 3.1. nastawienie na rozwój i nastawienie na trwałość
- 3.2. Kanban zaaplikowany do nauki
- 3.3. "pożądane trudności" kluczem do efektywnej nauki
- 3.4. "celowe ćwiczenia" kluczem do budowanie umiejętności eksperckich
- 3.5. harmonogram twórcy i głęboka praca
- 3.6. nowicjusze, eksperci i "najlepsze praktyki"
- 3.7. Dreyfus Squared: konfiguracje pracy w parach

4. Decyzje

- 4.1. błędy poznawcze w inżynierii oprogramowania
- 4.2. systematyczne radzenie sobie z błędami - framework WRAP
- 4.3. poszerz wachlarz możliwości (np. multitracking)
- 4.4. skonfrontuj założenia z rzeczywistością (np. adwokat diabła)

4.5. nabierz dystansu (np. reguła 10/10/10)

4.6. przygotuj się na błędy (np. premortem)

5. Cele

5.1. nadrzędny cel inżynierii oprogramowania

5.2. produkty zamiast projektów (#noprojects)

5.3. mityczne "business value" oraz impakt biznesowy

5.4. "move fast and don't break things" - Continuous Delivery wersja matematyczna

5.5. co warto mierzyć, a czego nie warto mierzyć

5.6. zespół zorientowany na wynik, a nie na aktywność oraz mit zajętości

5.7. utrzymać wysoką motywację: reguła postępu, szympansy i pszczoły