



Wykłady z inżynierii oprogramowania. Zarządzanie projektem informatycznym.

Zawartość

| | | |
|-----|--|---|
| 1. | Projekt informatyczny..... | 1 |
| 2. | Zarządzanie procesami wytwarzania oprogramowania | 2 |
| 3. | Zarządzanie ryzykiem (risk management)..... | 3 |
| 3.1 | Analiza ryzyka projektu..... | 4 |
| 3.2 | Plan zarządzania ryzykiem projektu | 5 |
| 3.3 | Obliczanie ryzyka projektu..... | 5 |
| 4. | Role w projekcie informatycznym..... | 7 |
| 4.1 | Kierownik Projektu (PM Agile) | 7 |
| 4.2 | Release Manager..... | 7 |
| 4.3 | Test Manager | 8 |
| 4.4 | Quality Engineer | 8 |
| 4.5 | Security Engineer..... | 9 |
| 4.6 | Inżynier ds. testów automatycznych | 9 |

1. Projekt informatyczny

Projekt informatyczny - jest to zbiór przedsięwzięć prowadzących do powstania programu komputerowego.

Częściami projektu informatycznego są

- procesy wytwarzania oprogramowania,
- procesy zarządzania związane z wytwarzaniem oprogramowania.

Procesy wytwarzania oprogramowania obejmują realizację zadań określonych we wszystkich fazach wytwarzania oprogramowania.

Procesy zarządzania związane z wytwarzaniem oprogramowania

- zarządzanie projektem
 - zarządzanie zasobami ludzkimi, w tym szkolenia
 - zarządzanie finansami/budżetem projektu,
 - zarządzanie ryzykiem,
 - zarządzanie procesem wytwarzania oprogramowania, planowanie prac związanych z wytwarzaniem oprogramowania, kontrola terminowości wykonywania zadań (kontrolę terminowości realizacji celów pośrednich milestones),
 - kontrola podwykonawców,
- zarządzanie jakością,
- zarządzanie konfiguracją i zmianami,
- zarządzanie testami,
- zarządzanie problemami.

Bibliografia

- I. Sommerville, Inżynieria oprogramowania, PWN, 2020
- R. S. Pressman, Software Engineering A Practitioner's Approach, McGraw-Hill, 2010.
- B. Rumpe, Agile Modeling with UML, Springer, 2017.
- K. Schwaber, Agile Project Management with Scrum, Microsoft Press, 2004

2. Zarządzanie procesami wytwarzania oprogramowania

Dla każdego etapu wytwarzania oprogramowania należy zdefiniować

- podprocesy wytwarzania oprogramowania i zależności między nimi,
- zadania danego etapu,
- produkty powstające na danym etapie,
- zasoby potrzebne do realizacji zadań,
- zasoby ludzkie potrzebne do realizacji zadań,
- zakresy odpowiedzialności (role) uczestników na danym etapie projektu,
- przypisanie zasobów do każdego zadania,
- osoby odpowiedzialne za dany produkt (właściciel produktu),
- zasady kontroli i akceptacji produktów danego etapu,
- harmonogram danego etapu.

Proces wytwarzania oprogramowania można podzielić na etapy

- zbieranie, formułowanie wymagań,
- analiza funkcjonalna wymagań,
- analiza systemowa,
- projektowanie,
- implementacja (kodowanie), unit testy,
- instalacja, integracja, konfiguracja,
- testy (integracyjne, funkcjonalne, użytkownika).

W projekcie należy określić jakie dokumenty powstają

- na każdym etapie wytwarzania oprogramowania,
- w procesie zarządzania projektem.

Należy sporządzić

- zarządzenia i procedury organizacyjne dotyczące projektu,
- szablony planów projektu (plan projektu, plan zarządzania jakością, ryzykiem, konfiguracją, testami, wdrożeniem),
- szablony dokumentów powstających w procesie wytwarzania oprogramowania (kontrakt, specyfikacja i analiza wymagań, use case'y, projekt systemu, opis wersji systemu, dokumentacja użytkownika,
- raporty,
- formularze kontrolne.

10 elementów wpływających na sukces projektu

1. Istnienie specyfikacji produktu.
2. Istnienie szczegółowego prototypu 'interfejsu użytkownika'.
3. Istnienie realistycznego harmonogramu projektu.
4. Szczegółowe i precyzyjne sformułowanie priorytetów projektu.
5. Aktywne zarządzanie ryzykiem.
6. Istnienie planu zapewnienia jakości dla projektu.
7. Szczegółowe określenie ról, zakresu obowiązków i czynności w projekcie.
8. Realizowanie planu zarządzania konfiguracją.
9. Określenie architektury produktu.
10. Istnienie planu integracji i wdrożenia.

9 elementów wpływających na porażkę projektu

1. Brak planów projektu.
2. Tworzenie nierealistycznych planów projektu.
3. Brak analizy ryzyka projektu (brak planu zarządzania ryzykiem).
4. Stosowanie tego samego planu do wszystkich projektów.
5. Stosowanie standardowych planów bez ich adaptacji do konkretnego projektu.
6. Brak aktualizowania planów w trakcie trwania projektu.
7. Planowanie nieistotnych szczegółów dla danego etapu projektu (planowanie zbyt wielu elementów).
8. Planowanie z uwzględnieniem wzrostu efektywności pracy.
9. Brak analizy błędów popełnionych w procesie planowania.

3. Zarządzanie ryzykiem (risk management)

Ryzykiem projektu - jest potencjalne zdarzenie stwarzające zagrożenie dla realizacji celów projektu.

Zarządzanie Ryzykiem - jest ciągłym procesem identyfikowania, analizy i kontroli zdarzeń zachodzących w projekcie informatycznym pod kątem potencjalnego negatywnego ich wpływu na cele projektu, spowodowania szkody w projekcie.

Proces zarządzania ryzykiem określony jest przez

- procedury planowania zarządzania ryzykiem,
- procedury identyfikacji ryzyka w projekcie,
- procedury analizy ryzyka,
- zasad definiowania metryk ryzyka,
- zasad określania prawdopodobieństwa i wag ryzyka,
- zasad podejmowania decyzji związanych z ryzykiem w projekcie,
- procedury monitorowania realizacji prac związanych z usunięciem ryzyka lub minimalizacji szkód ryzyka,
- procedury dokumentowania procesu zarządzania ryzykiem,
- określeniu procedur udoskonalania procedur zarządzania ryzykiem.

Planowanie procesu zarządzania ryzykiem polega na

- określeniu obszarów problemowych które będą podlegały procesowi zarządzania ryzykiem,
- określeniu celów zarządzania ryzykiem i odniesienie ich do celów projektu,
- stworzeniu planu zarządzania ryzykiem,
- odniesienia planu zarządzania ryzykiem do innych planów (plan projektu, plan zarządzania jakością).

3.1 Analiza ryzyka projektu

Prawdopodobieństwem ryzyka - jest prawdopodobieństwo zajścia niepożądanego zdarzenia w projekcie.

Waga ryzyka (wielkość szkody) - jest kosztem, miarą szkody jaka zajście ryzyka spowoduje w projekcie.

Stopień ryzyka - jest iloczyn prawdopodobieństwa ryzyka i wagi ryzyka (wielkości szkody).

$$\text{Stopień Ryzyka} = \text{Prawdopodobieństwo Ryzyka} * \text{Waga ryzyka}$$

Celem analizy ryzyka jest określenie

- prawdopodobieństwa zajścia zidentyfikowanego ryzyka,
- wagi ryzyka (oszacowanie wielkości szkody jakie może spowodować zajście ryzyka),
- metod usunięcia ryzyka (niedopuszczenie do zajścia ryzyka),
- metod minimalizacji szkód w sytuacji zajścia ryzyka,
- kosztów usunięcia, minimalizacji, przeniesienia ryzyka,
- zasad monitorowania ryzyka (okresowe szacowanie prawdopodobieństwa i wagi ryzyka).

inne cele

- identyfikacja zagrożeń w procesach zarządzania projektem i w procesach wytwarzania oprogramowania,
- ustalenie priorytetów działań w oparciu o ustalone stopnie ryzyka,
- dostarczanie danych do procesu zarządzania jakością.

Analiza ryzyka jest procesem polegającym na

- określaniu prawdopodobieństwa zajścia ryzyka,
- określaniu wagi ryzyka,
- identyfikowaniu czynności minimalizujących prawdopodobieństwo zajścia ryzyka,
- identyfikowaniu czynności minimalizujących skutki zajścia ryzyka (minimalizacja wielkości szkody),
- przypisaniu wartości prawdopodobieństwa zajścia ryzyka z podjętymi działaniami minimalizującymi,
- określeniu wielkości szkody ryzyka z podjętymi działaniami minimalizującymi,
- identyfikowaniu skutków zajścia ryzyka (szkód),
- określaniu prawdopodobieństwa wystąpienia szkody,
- szacowaniu wielkości każdej szkody
- określeniu zasad monitorowania ryzyka.

Przed przystąpieniem do analizy ryzyka należy określić **rodzaje ryzyka**.

Rodzaj ryzyka określany w zależności od typu projektu.

Przykład 1. Rodzaje ryzyka: ryzyko projektu, ryzyko funkcjonalne.

3.2 Plan zarządzania ryzykiem projektu

Plan zarządzania ryzykiem projektu jest dokumentem kontrolnym, określającym jak procedury zarządzania ryzykiem identyfikacja ryzyka, analiza i ocena ryzyka, kontrola ryzyka, będą stosowane w danym projekcie.

Plan zarządzania ryzykiem projektu określa

- relacje procesu zarządzania ryzykiem do innych procesów w projekcie,
- odniesienie celów zarządzania ryzykiem do celów projektu,
- metody realizacji celów zarządzania ryzykiem w projekcie,
- harmonogram prac związanych z zarządzaniem ryzykiem w projekcie,
- dokumentację procesu zarządzania ryzykiem w projekcie.

Elementy planu zarządzania ryzykiem projektu

1. Zakres dokumentu
2. Ogólne zasady zarządzania ryzykiem projektu
3. Organizacja procesu zarządzania ryzykiem
 - 3.1. Role w procesie zarządzania ryzykiem
 - 3.2. Zakres odpowiedzialności dla ról
 - 3.4. Opis procesu i procedur zarządzania ryzykiem
4. Ocena ryzyka
 - 4.1. Identyfikacja ryzyka
 - 4.1.1. Referencje do formularzy kontrolnych*
 - 4.1.2. Terminologia
 - 4.2. Analiza ryzyka
5. Minimalizacja ryzyka
6. Kontrola ryzyka
 - 6.1. Zasady kontroli ryzyka, definicja miar ryzyka
 - 6.2. Metody pozyskiwania ryzyk
 - 6.3. Harmonogram kontroli i raportowania
7. Komunikacja, obieg dokumentów w procesie RM
8. Raportowanie analizy ryzyka
9. Dokumentacja ryzyka
 - 9.1. Lista dokumentów zarządzania ryzykiem
 - 9.2. Zasady kontroli wersji i dostępu do dokumentów.

* formularz kontrolny (ang. checklist) zbiór formularzy służących analizie ryzyka (identyfikowaniu, szacowaniu ryzyka).

Każdy proces zarządzania w projekcie wymaga analizy ryzyka

- analiza ryzyka we wszystkich fazach procesu wytwarzania oprogramowania (w fazie analizy i projektowania, implementacji, testów, instalacji),
- analiza ryzyka w procesie zarządzania konfiguracją,
- analiza ryzyka w procesie zarządzania jakością.
- analiza ryzyka w procesie zarządzania testami.

3.3 Obliczanie ryzyka projektu

Czynności związane z szacowaniem ryzyka w projekcie

- identyfikacja i opis ryzyka,
- określenie **wagi ryzyka** (wielkości szkody ryzyka),
- przypisanie wartości **prawdopodobieństwa zajścia ryzyka**,
- identyfikacja i opis **czynności minimalizujące prawdopodobieństwa zajścia ryzyka**,
- identyfikacja i opis **czynności minimalizujące skutki zajścia ryzyka** (wielkości szkody),

- określenia kosztów działań minimalizujących,
- określenie wartości **prawdopodobieństwa zajścia ryzyka** z podjętymi działaniami minimalizującymi,
- określenie wielkości szkody z podjętymi działaniami minimalizującymi,
- identyfikacja **szkód** w przypadku zajścia ryzyka,
- identyfikacja i opis **czynności likwidujących skutki** (likwidacja szkód) zajścia ryzyka,
- określenia kosztów usuwania szkód,
- określenie **zasad monitorowania ryzyka**.

Możliwe wartości 'prawdopodobieństwa ryzyka' i 'wielkość szkody ryzyka' określone są w "Planie zarządzania ryzykiem projektu".

Prawdopodobieństwo ryzyka - prawdopodobieństwo zajścia niekorzystnego zjawiska.

- Bardzo mało prawdopodobne (25%), wartość: 1.
 Mało prawdopodobne (50%), wartość: 2.
 Prawdopodobne (75%), wartość: 3.
 Bardzo prawdopodobne (100%), wartość: 4.

W skali 1-8, waga ryzyka określa przewidywana wielkość szkody, jaką może spowodować zajście niekorzystnego zdarzenia.

- Mała szkoda, wartość: 1.
 Średnia szkoda, wartość: 2.
 Poważna szkoda, wartość: 4.
 Krytyczna szkoda, wartość: 8.

Przykład 1. Tabela analizy ryzyka projektu.

| Nr | Rodzaj ryzyka | Waga (A) (1-16) | Prawdopodobieństwo ryzyka (B) (1-4) | Stopień ryzyka (A*B) | Działania minimalizujące ryzyko | Waga (A) (1-16) | Prawdopodobieństwo ryzyka (B) (1-4) | Stopień ryzyka (A*B) |
|-----|---|-----------------|-------------------------------------|----------------------|---------------------------------|-----------------|-------------------------------------|----------------------|
| 1 | Ryzyko projektu | | | | | | | |
| 1.1 | Pakiety instalacyjne będą dostarczone z opóźnieniem. | 1 | 4 | 4 | | 1 | 2 | 2 |
| | | | | | | 0 | 4 | 0 |
| 1.2 | Dokumentacja instalacyjna nie będzie dostarczona z pakietami. | 4 | 4 | 16 | | 3 | 4 | 12 |
| 1.3 | Procedury archiwizacyjne i reinstalacyjne systemu nie będą gotowe na czas testów. | 8 | 4 | 32 | | 7 | 4 | 28 |
| 1.4 | Znalezione błędy podczas testów nie będą usuwane zgodnie z harmonogramem testów. | 4 | 4 | 16 | | 4 | 3 | 12 |
| 2 | Ryzyko funkcjonalne | | | | | | | |
| 2.1 | Brak ostatecznych kryteriów akceptacji systemu. | 8 | 1 | 8 | | | | |
| 2.2 | Brak definicji konfiguracji systemu do testów. | 4 | 4 | 16 | | | | |

4. Role w projekcie informatycznym

4.1 Kierownik Projektu (PM Agile)

Zakres obowiązków

- kierowanie projektami IT od pomysłu do fazy stabilizacji, poprzez analizę, tworzenie, testy i wdrożenie
- planowanie i usprawnianie codziennych spotkań stand-up, regularne retrospekcje i prowadzenie międzypespółowych spotkań planistycznych
- wspieranie zespołu w zakresie samodzielnego zarządzania swoimi iteracjami
- rozwiązywanie problemów, tworzenie interakcji, które pomagają usprawniać współpracę
- nadzorowanie zgodności działań ze sprawnym framework'iem Scrum
- koordynacja działań, usuwanie barier w celu pomyślnej dostawy/realizacji procesu wytwarzania produktów.

Wymagania

- min. 5 lat doświadczenia zawodowego jako Project Manager lub Scrum Master
- min. 3 lata doświadczenia w zarządzaniu zespołem projektowym realizującym zadania w obszarze tworzenia systemów informatycznych
- znajomość zwinnych metod prowadzenia projektów (Scrum, Crystal, DSDM, FDD) oraz tradycyjnych metod zarządzania projektami (Waterfall)
- znajomość narzędzi MS Project, JIRA, Confluence
- zrozumienie procesu wytwarzania oprogramowania

Mile widziane

- praktyczne doświadczenie w zakresie framework'u i/lub certyfikacja AgilePM® Foundation lub Practitioner i/lub certyfikacja SAFe® Product Owner/Product Manager
- certyfikacja PRINCE2® Foundation lub Practitioner
- doświadczenie w zakresie udoskonalania procesów informatycznych i/lub oprogramowania.

4.2 Release Manager

Zakres obowiązków

- planowanie i koordynacja prac dotyczących wdrożeń rozwiązań informatycznych na środowiskach testowych i produkcyjnych
- budowa oraz obsługa procesów zarządzania wydaniem
- współpraca z perspektywy wdrożeń z wieloma zespołami dostarczającymi rozwiązania informatyczne zarówno zwinnie (Agile), jak i klasycznie (kaskadowo)

Wymagania

- co najmniej 2-3 letnie doświadczenie w zarządzaniu wydaniem jako Release Manager
- znajomość ITIL w szczególności te dotyczące Release management, Change management
- doświadczenie w SCRUM i rozumienie procesów wytwórczych SDLC (Software Development Life Cycle)
- znajomość narzędzi JIRA i Confluence.

Jira Software - to narzędzie przeznaczone dla członków zespołu programistów, które służy do planowania, śledzenia procesu tworzenia i wdrażania oprogramowania wysokiej jakości.

Confluence - to narzędzie służące do zarządzania zespołem, dokumentacją projektu.

Release Manager 2.

Zakres obowiązków

- planowanie i zarządzanie procesem wdrażania zmian w systemach IT
- dbanie o rozwój i usprawnienia procesów z obszaru Release Management
- koordynacja prac dotyczących przygotowywania wdrożeń, harmonogramów oraz innych produktów związanych z wydaniem zmian biznesowych, infrastruktury IT oraz Bezpieczeństwa
- zarządzanie środowiskami nieprodukcyjnymi (Test, PrePROD)

- rozwój narzędzi związanych z procesem Release Management (JIRA, Jenkins) z ukierunkowaniem na CI/CD
- komunikowanie do organizacji i obszarów biznesowych informacji nt. harmonogramu wdrożeń
- odpowiedzialność za dokumentację procesową na potrzeby audytowe lub raportowe
- ścisła współpraca z menedżerami wyższego szczebla, właścicielami aplikacji i zespołami rozwojowymi, kierownikami projektów, użytkownikami biznesowymi w zakresie wydań, zakresu, koordynacji i śledzenia postępów
- bliska współpraca z zespołami Rozwoju Oprogramowania, Infrastruktury i QA

Wymagania

- dobra znajomość procesu Release Management oraz Change Management (ITIL)
- znajomość narzędzi wspierających CI/CD (Jenkins, Bamboo)
- znajomość narzędzi do kontroli wersji np. GIT
- znajomość zasad korzystania z systemów kontroli wersji (branchowanie, mergowanie, itd.)
- doświadczenie w pracy z narzędziami Continuous integration (CI) i Continuous Delivery (CD)

4.3 Test Manager

Responsibilities

- accountable for ensuring the quality is built into development of all solutions at all stages within the development lifecycle
- define and develop projects' test strategies and plans and associated deliverables
- plan and schedule project timelines and milestones using appropriate tools
- coach, mentor, motivate and supervise team members. Provide expert practice advice and guidance to junior colleagues.
- develop test capabilities and increase knowledge and understanding of the importance of building in quality across the wider development community

Skills and attributes

- significant experience in software testing, preferably in financial area.
- proven team leadership experience in quality assurance (both online and offline)
- strong knowledge of software QA methodologies, tools (defect tracking and test management) and processes
- ability to plan, track and report the status of the deliverables
- experience in working in Agile methodologies

Your background:

5+ experience as a QA Lead, Test Lead or Test Manager in large, international projects,

Ideally, you'll also have:

- experience with performance, automation and/or security testing,
- testing/QA certifications such as ISTQB Certified Tester,
- technical knowledge in programming, scripting and related standards (Python, bash, Perl, Java, XML, JavaScript, HTML etc.)

4.4 Quality Engineer

Your responsibilities

- recognizing and classifying the typical risks associated with the performance, security, reliability, portability and maintainability of software systems.
- providing technical elements to the planning, design and execution of tests for mitigating performance, security, reliability, portability and maintainability risks.
- selecting and applying appropriate test techniques to ensure that tests provide an adequate level of confidence, based on design coverage.
- effectively participating in reviews with developers and software architects applying knowledge of typical defects in the code and architecture.
- improving the quality characteristics of code and architecture by making use of different analysis techniques.
- work with developers to train them in effective testing techniques (including exploratory testing).
- selecting appropriate tools to automate technical testing tasks.

- understanding the technical issues and concepts in applying test automation.
- creating test plans / strategies and participating in creating test documentation, test scenarios.

Our requirements

- we are looking for a person who has experience in assessing and mitigating risk, resolving quality issues, and implementing changes to improve engineering efficiency.
- good knowledge of issues and tools related to application testing is essential for this position.
- familiarity with various forms of testing (unit, integration, acceptance, exploratory).
- ability to define project or team quality objectives and measure them.

Optional

- ISTQB Advanced Level Technical Test Analyst certificates.
- experience testing projects on Magento 2 (or comparable e-commerce platform).
- knowledge of the following technologies: Javascript, Nuxt, php, Headless, Magento.

4.5 Security Engineer

Our requirements

- minimum 2 years of experience on security positions related to onboarding, maintaining or creating security systems,
- know-How related to most important cyber security dangers, including attack vectors and functionality of cybercriminal methods,
- knowledge of firewalling solutions (Checkpoint, Paloalto, Cisco, Fortinet),
- basic of cryptographic (PKI infrastructure, Protocols, Ciphers),
- experience in onboarding security tools in the IT environments,
- knowledge of cloud solutions (Microsoft Azure, Amazon Web Services, Google Cloud Platform),

Optional

- know-how of issues and methods of forensic investigations analysis,
- security certificates as Security+, CISSP (Certified Information Systems Security Professional) or related,
- knowledge of ISO 27001 standard,
- working with CIS (Certification Information Security), National Institute of Standards and Technology standards,
- working in ITIL oriented environment.

4.6 Inżynier ds. testów automatycznych

Twój zakres obowiązków

- Strukturyzacja, projektowanie i planowanie scenariuszy oraz przypadków testowych
- Przeprowadzanie testów integracyjnych
- Dbanie o jakość wykonywanych zadań
- Raportowanie i analiza błędów
- Współpraca z klientem i zespołem projektowym przy rozwoju i utrzymaniu projektu

Wymagania

- Co najmniej 1 rok doświadczenia w testowaniu integracyjnym (REST API, SOAP API)
- Umiejętność pisania testów automatycznych
- Znajomość narzędzi: Postman, SoapUi, JIRA
- Rozumienie procesu testowego i jego elementów

Mile widziane

- Praca w projektach zwinnych (Agile/Scrum)
- Wiedza z zakresu metodyki testów zgodnej ze standardem ISTQB
- Doświadczenie w pracy z systemem kontroli wersji np. GIT
- Umiejętność wykonywania testów niefunkcjonalnych (wydajność, bezpieczeństwo)

Tester Oprogramowania

Twój zakres obowiązków

- projektowanie, dokumentowanie i realizacja testów funkcjonalnych i нефункциональных oprogramowania komputerowego w zależności od potrzeb projektowych.
- konsultowanie i dobór rozwiązań technologicznych w ramach projektu.
- wycena czasu pracy i przygotowywanie estymacji testów.
- middle+/Senior: projektowanie i wdrażanie strategii testowych w zakresie testów funkcjonalnych oraz нефункциональных.

Wymagania

- minimum 3 lata doświadczenia w testowaniu oprogramowania.
- znajomość techniki projektowania przypadków testowych.
- umiejętność zgłaszania błędów i tworzenia procedury testowe.
- doświadczenie w przygotowywaniu dokumentacji testowej.
- znajomość podstawowych procesów testowych, podstawowych narzędzi i platform testowych.

Performance Test Specialist

Twój zakres obowiązków

- udział w opracowaniu oraz we wdrożeniu procesu testów wydajnościowych w projekcie
- tworzenie testów wydajnościowych - JMeter
- propagowanie oraz rozwijanie wiedzy eksperckiej z dziedziny testów wydajnościowych
- tworzenie oprogramowania w technologii .NET, wykorzystywanego dla realizowanych testów wydajnościowych
- praca w pełnym, dojrzałym cyklu tworzenia oprogramowania: kontrola jakości kodu, Continuous Integration/Deployment, wersjonowanie aplikacji, zautomatyzowane wdrożenia, wiele środowisk testowych, automatyzacja testów na maszynach wirtualnych
- organizowanie oraz przeprowadzanie testów wydajnościowych
- przeprowadzanie testów automatycznych funkcjonalnych

Nasze wymagania

- 2 lata doświadczenia w automatyzacji testów i testowaniu oprogramowania
- co najmniej rok doświadczenia w tworzeniu testów wydajnościowych
- praktyczną znajomość co najmniej jednego z narzędzi do testów wydajnościowych
- doświadczenie w programowaniu w C#, JavaScript (nie tylko związane z testowaniem)