

Respondent: **Varvara Shafarevich** (Grupy: L7, L7-Delta, Test, Test-poprawkowy) Przesłane w dniu: niedziela, 25 lutego 2024, 10:44

Formularz samooceny zespołu - Sprint #1

1 * Identyfikator zespołu (np. I1-1-Alpha)

L7-Poprawa

2 * Osoby w zespole (1 osoba w linii)

Varvara Shafarevich 153932 Stanisław Borowiec 150775

Definition of Done

3 * 1. Kod jest pod kontrolą systemu zarządzania wersjami (Github) i prowadzący ma do niego dostęp

☐ 0 - Nie

☒ 20 - Tak

4 * Proszę podać URL do repozytorium na GitHub.

<https://github.com/stanislawborowiec/sortingmadness>

5 * 2. Kod określający inkrement produktu jest oznaczony w repozytorium (tag = numer wersji) oraz odpowiednia wersja w pom.xml

☐ 0 - Nie

☒ 20 - Tak

6 * Tag oznaczający oznaczoną wersję na koniec sprintu.

0.4

7 * 3. Aplikacja buduje się w sposób automatyczny (kompilacja, stworzenie paczki do dystrybucji - np. jar)

☐ 30 - Tak

☒ 0 - Nie

8 * 4. Skonfigurowano środowisko ciągłej integracji i prowadzący ma do niego dostęp (serwer CI reaguje na zmiany w repozytorium kodu i uruchamia proces budowania)

☒ 0 - Nie

☐ 30 - Tak

(9) * Proszę podać URL do systemu ciągłej integracji

10 * 5. Dokumentacja kodu (<http://www.oracle.com/technetwork/articles/java/index-137868.html>)
Dokumentacja generuje się w sposób automatyczny podczas budowy oprogramowania

☐ 0 - Nie

☒ 10 -Tak

11 * 5. Dokumentacja kodu (<http://www.oracle.com/technetwork/articles/java/index-137868.html>)
Liczba klas / interfejsów posiadających dokumentację.

☐ 0 - żadna

☐ 30 - Przynajmniej 1 klasa / interfejs posiada pełną dokumentację techniczną* (javadoc, itp.)

☒ 40 - Przynajmniej 3 klasy / interfejsy posiadają pełną dokumentację techniczną* (javadoc, itp.)

☐ 50 - (tylko grupy 5 osobowe) Przynajmniej 5 klasy / interfejsy posiadają pełną dokumentację techniczną* (javadoc, itp.)

12 * Skonfigurowano środowisko do zarządzania rejestrem produktu i rejestrem sprintu, prowadzący ma do niego dostęp (np. Github Issues, Github Projects, Trello)

- ☐ 0 - Nie
☒ 20 - Tak

13 * URL do środowiska do zarządzania rejestrem produktu / sprintu (np. Github Issues, Github Projects, Trello)

<https://trello.com/b/cVtVzi0o/sorting-madness>

14 * 7. Analizowalność kodu (logowanie wykonania) - W programie użyto biblioteki SL4J do logowania wykonania na poziomie DEBUG i INFO

- ☒ 0 - Nie
☐ 10 - Tak

15 * 8. Zaprojektowano model klas (dla własnych klas, jeśli więcej niż 10 to ograniczyć do najważniejszych klas) z wykorzystaniem diagramu klas UML i umieszczono go w repozytorium lub na wiki projektu

- ☐ 0 - Nie
☒ 40 - Tak

16 * URL do diagramu klas (plik graficzny)

https://github.com/stanislawborowiec/sortingmadness/blob/main/Diagram_uml.png

17 * 8. Opracowano diagram sekwencji obrazujący sekwencje przetwarzania wybranej operacji począwszy od otrzymania żądania REST do zwrócenia wartości w kontrolerze REST

- ☒ 0 - Nie
☐ 30 - Tak (tylko grupa 5 osobowa)

(18) * URL do diagramu sekwencji w postaci pliku graficznego

19 * 9. Wzorce projektowe - Podczas implementacji wykorzystano sugerowany wzorec projektowy w proponowanym zakresie: TextTransformer - Dekorator wykorzystany do dynamicznego tworzenia obiektu transformującego; Scenario Quality Checker - Wizytator będzie użyty do analizy scenariuszy i pod-scenariuszy; Building Info - Kompozyt do reprezentacji struktury pomieszczeń oraz wywoływania operacji mających bezpośredni związek ze strukturą danych (obliczenia powierzchni, kubatury, zużycia energii, mocy oświetlenia). Funkcje raportujące mogą być zaimplementowane jako Wizytator, JSON tools - wzorec dekorator do dynamicznego tworzenia wyjścia na bazie poszczególnych opcji (np. filtrowanie, minifikacja), SortingMadness - wzorec strategii dla poszczególnych algorytmów sortowania i zbierania czasu wykonania.

- ☐ 0 - Nie
☒ 40 - Tak

Prowadzenie projektu

20 * 1. Wprowadzono rejestr produktu do narzędzia zarządzania (np. Trello)

- ☐ 0 - Nie
☒ 20 - Tak

21 * 2. Rejestr sprintu - Przeniesiono do rejestru sprintu elementy rej. produktu, które mają być zrealizowane w sprincie

- ☐ 0 - Nie
☒ 10 - Tak

22 * 2. Rejestr sprintu - Dodano zadania do poszczególnych elementów rej. produktu realizowanych w sprincie

- ☐ 0 - Nie
☒ 30 - Tak

Wartość biznesowa

23 * Punkty za wartość biznesową - uwaga wszystkie elementy Definition of Done oznaczone jako **"Czy minimum?" na TAK muszą być spełnione!**

[Link do zasad punktowania](#)

- ☐ 0 - przynajmniej jeden element DoD oznaczony jako minimalny nie został spełniony lub nie udało się zaimplementować żadnego elementu Product Backlogu
- ☐ 50 - Inkrement zawiera implementację elementów rejestru produktu o wartości 2 BV
- ☒ 80 - Inkrement zawiera implementację elementów rejestru produktu o wartości 4 BV
- ☐ 110 - (tylko grupy 5 os.) Inkrement zawiera implementację elementów rejestru produktu o wartości 5 BV
- ☐ 140 - (tylko grupy 5 os.) Inkrement zawiera implementację elementów rejestru produktu o wartości 6 BV lub więcej

Podsumowanie

24 * Zdobyta liczba punktów (według samooceny zespołu)

330

25 * Podział % punktów w zespole (dzielimy % a nie konkretne punkty).

Stanisław Borowiec 50% Varvara Shafarevich 50%