

Projekt Zespołowy Projekt

Paweł Rzążewski

20.02.2017

Informacje ogólne

- Podzielony na części
- Regularna kontrola postępów pracy
- Oceniany jest nie tylko finalny projekt, ale też (a nawet przede wszystkim) sposób jego realizacji

Podział na zespoły

- Zespoły 3-4 osobowe
- Warto zadbać o zróżnicowanie każdego zespołu (by w każdym zespole znalazła się osoba biegła w programowaniu, osoba pisząca dobre dokumentacje, osoba dobra w pilnowaniu harmonogramów itp.)
- Każdy zespół wybiera lidera. Zadaniem lidera jest kontrola pracy całego zespołu.
- Za każdą z części projektu odpowiada cały zespół (co nie znaczy, że każdy dostanie identyczną ocenę końcową).

Terminarz projektu

27.02.2017	Podział na zespoły, wybór lidera zespołu
03.03.2017	Szczegóły projektu uzgodnione między liderami
10.03.2017	Harmonogram pracy nad projektem
13.03.2017	Dokumentacja funkcjonalna
03.04.2017	Projekt techniczny
08.05.2017	Kontrola postępów - działające części aplikacji
22.05.2017	Wersja beta aplikacji
04.01.2017	Aplikacja
05.06.2017	Raport z testów aplikacji
05.06.2017	Finalny harmonogram
12.01.2017	Prezentacja projektów

+ regularne spotkania liderów z prowadzącym i sobą nawzajem

Punktacja

10%	harmonogram wstępny
15%	dokumentacja funkcjonalna
25%	projekt techniczny
40%	aplikacja (w tym regularne postępy i prezentacja)
10%	raport z testów
100%	razem

Punktacja - cd.

- Warunkiem koniecznym zaliczenia jest oddanie wszystkich części projektu
- Zaliczenie projektu: ponad 50 pkt z projektu
- Spóźnienie jest karane -10 pkt za każdy rozpoczęty tydzień.

Harmonogram:

- Godziny pracy
- Kamienie milowe z terminami
- Wstępnie wydzielone zadania
- Wstępny podział pracy

Dokumentacja funkcjonalna:

- Opis problemu
- Algorytmy używane w rozwiązaniu
- Wymagania aplikacji - funkcjonalne
- Wymagania aplikacji - нефункционалне
- Przypadki użycia wraz z sytuacjami błędnymi
- Ewentualne zmiany w harmonogramie (z objaśnieniem przyczyn)

Projekt techniczny:

- Opis rozwiązania problemu
- Projekt klas (diagram UML)
- Szczegółowy opis używanych formatów plików
- Ewentualne zmiany w harmonogramie (z objaśnieniem przyczyn)

Artefakty

Aplikacja:

- Zgodność z założeniami
- Styl kodowania
- Testy jednostkowe
- Dokumentacja kodu
- Współpraca z aplikacjami innych grup

Raport z testów:

- Szczegółowy opis testów
- Testy poprawności modułów, testy integracyjne
- Testy szybkości działania
- Wnioski z testów

Artefakty

Dodatkowo:

- Ostateczny podział pracy
- Używane narzędzia
- Jak przebiegała praca?
- Ostateczny harmonogram

Problem poligrafa

- drukuje różne ulotki, wizytówki (ogólnie nazywane użytkami) na dużych arkuszach i potem wycina przy użyciu maszyny
- maszyna tnie cały arkusz, nie można wycinać ze środka
- można wycinać wiele identycznych arkuszy naraz
- po rozcięciu arkusza, jego części mogą być nadal wycinane
- mamy dane zamówienia – rozmiary i liczbę sztuk (np. 1000 ulotek w rozmiarze 10 x 15 cm)
- chcemy rozmieścić nasze użytki na arkuszu w taki sposób, aby:
 - dało się je wyciąć za pomocą naszej maszyny
 - każde zamówienie zostało zrealizowane
 - zminimalizować liczbę użytych arkuszy

Typowe dane

Typowe rozmiary arkuszy to:

- SRA3 (450 x 320 mm)
- A1 (841 x 594 mm)
- A2 (594 x 420 mm)
- B2 (700 x 500 mm)

Typowe rozmiary użytków (netto; brutto) to:

- W1 (90 x 50 mm; 96 x 56 mm)
- W21 (85 x 55 mm; 91 x 61 mm)
- A5 (210 x 148 mm; 216 x 154 mm)
- A6 (148 x 105 mm; 154 x 111 mm)
- DL (210 x 100 mm; 216 x 106 mm)

Na przykład na arkuszu SRA3 mieści się 21 typowych wizytówek W1 (wiemy to z doświadczenia), ale co jeśli mamy bardziej skomplikowane zamówienie?

Zadanie

W ramach zadania należy przygotować:

- format danych wejściowych (wspólny dla wszystkich grup)
- format danych wyjściowych (wspólny dla wszystkich grup)
- aplikację pozwalającą wczytanie zamówienia i zwracającą liczbę arkuszy oraz rozmieszczenie poszczególnych użytkowników
- aplikacja musi też umożliwiać wczytanie pliku z rozwiązaniem i graficzne przedstawienie rozwiązania
- algorytmy