Wedding Planner

# Testowanie

## Opis metodyki testowania

## Uzyskiwane wyniki pomiarów

## Analiza i interpretacja uzyskanych wyników

# Podsumowanie

## Wykres wypalania

Na poniższym wykresie został przedstawiony proces spalania dla poszczególnych sprintów. Jak widać, udało nam się skończyć projekt mniejszym nakładem pracy niż było to estymowane. Przyczyną była dość znacząca powtarzalność niektórych fragmentów kodu, dzięki czemu łatwiej było napisać/przekształcić poprzedni fragment, jeśli robiło się przed chwilą coś bardzo podobnego.

Powyższy wykres został zrobiony na podstawie tabeli, która zawiera zsumowane wartości przypuszczalnej i rzeczywistej czasochłonności. Dane zostały zebrane dzięki wtyczce „Scrum for Trello” z tablicy Trello[[1]](#footnote-1).

|  |  |  |
| --- | --- | --- |
| Sprint | Consumed Points | Estimated Points |
| 1 | 8 | 8 |
| 2 | 5 | 5 |
| 3 | 6 | 6 |
| 4 | 9 | 9 |
| 5 | 8 | 11 |
| 6 | 8 | 8 |
| 7 | 5 | 4 |

## Wnioski

Przyznam, że to była pierwsza tego typu strona internetowa, którą rzeczywiście robiłam. Do tej pory moje prace dotyczyły praktycznie samego front-endu, albo jakiegoś wycinku np. same zapytania SQL. Jeszcze do niedawna wydawało mi się, że pisanie stron/aplikacji internetowych jest bardzo fajne i zajmujące. Teraz trochę zmienił mi się pogląd na tę działkę informatyki. Dość spora część zadań jest powtarzalna – na wielu podstronach trzeba było dodać bardzo podobny back-end zajmujący się np. dodawaniem nowych rekordów do bazy. Jednak nie można było użyć tego samego kodu, bo istniało kilka istotnych różnic.

Mimo to, mogę z pewnością stwierdzić, że środowisko Visual Studio z opensourcowym frameworkiem ASP.NET, biblioteką jQuery i backendem zrobionym w C# jest stosunkowo przyjemne dla osób początkujących. W Internecie znajduje się bardzo dużo tutoriali i odpowiedzi na problemy, które pojawiają się podczas kodowania. Największą barierą do pokonania było dopasowanie odpowiednich elementów do siebie, tzn. dla widoków z rozszerzeniem aspx trzeba było znaleźć jak najprostszy sposób na zrobienie bazy danych oraz logiki, która będzie w stanie połączyć to wszystko. Znajdowanych propozycji było naprawdę wiele, ale w przypadku wybrania MVC trzeba by w jakiś sprytny sposób wrzucać uzyskiwane tabele do gotowych już widoków. W tym przypadku pracy byłoby dużo więcej, a efekty wizualne byłyby najprawdopodobniej gorsze.

Najprostszym elementem było zrobienie tabel i bazy danych – pewnie wynika to z mojego doświadczenia w tej dziedzinie. Natomiast pewnym zaskoczeniem dla mnie była elegancja wyglądu strony startowej zaraz po utworzeniu projektu. Zdziwiło mnie, że Visual Studio w tym przypadku proponuje naprawdę wiele praktycznie gotowych już „półproduktów”, jak m.in. kilka widoków, logika do logowania i rejestracji.

Gdy chodzi o język C# to widzę w nim bardzo wiele podobieństw do C++. Pod koniec pisania logiki praktycznie nie pamiętałam już, że jest to inny język programowania. W C# podoba mi się to, że łatwo napisać w nim coś porządniejszego.

Przechodząc do organizacji pracy, to szczególnie przydał mi się Git. Mogłam bez problemu na innym branchu zacząć coś robić i etapami commitować, a jeśli nie wyjdzie, to po prostu powrócić do poprzedniej wersji. Podoba mi się to, że mogę powrócić zarówno do poprzedniej wersji commita, jak i do tego co jest już „spushowane”.

Metodyka zwinna, a mianowicie tablica na Trello [[2]](#footnote-2) i sposób pracy nad projektem bardzo przydał się szczególnie na początku – przy planowaniu, ustalaniu wszystkich istotnych elementów dot. koncepcji projektu. Potem był wyznacznikiem postępów w pracy i wskazywał na to, ile jeszcze zostało do zrobienia. W Trello można oznaczyć inną osobę i ta osoba dostanie powiadomienie, widać na bieżąco, co jest w jakim stopniu skończone i bardzo istotne dla mnie jest to, że działa szybko (w przeciwieństwie do Jiry).

1. <https://trello.com/b/S0kYwbQ4/pai2017gargulce> [↑](#footnote-ref-1)
2. <https://trello.com/b/S0kYwbQ4/pai2017gargulce> [↑](#footnote-ref-2)