Ewa Gumińska 131686

Wojciech Stanisławski 143351

Krzysztof Świeczkowski 143365

Wojciech Żółtowski 143388

Projekt

**Użyteczność Oprogramowania**

### 1. Opis analizowanego rozwiązania

Projekt o nazwie “Narzędzie do śledzenia błędów oraz zarządzania projektami Scrum” jest realizowy w ramach projektu grupowego. Celem wytwarzanego systemu jest ułatwienie użytkownikom zarządzania projektami metodyką Scrum. Dodatkowo produkt będzie wspierał obsługę zgłoszeń błędów przez testerów. Chcemy stworzyć rozwiązanie idealne zarówno dla studentów, jak i młodych firm, darmowe, o otwarym kodzie źródłowym.

Projekt składać się będzie z serwera oraz klienta webowego uruchamianego w przeglądarce. Będzie eliminować większość wad innych produktów tego typu. Będzie mogła być uruchomiona na wielu różnych systemach operacyjnych, a także na platformach mobilnych. System tworzymy z myślą przede wszystkim o młodych biznesach. Liczymy na to, że dotrze on również do społeczności startupowej.

Klientem naszej aplikacji mogą być nie tylko zespoły informatyczne, ale także te prowadzące inną działalność niż informatyczną. Użytkownikami naszej aplikacji mogą być programiści, testerzy, Scrum masterzy, product ownerzy, a także inne osoby pracujące w metodyce Scrum.

**Kontekst użycia**

* nieduże projekty informatyczne,
* startupy,
* projekty studenckie,
* małe firmy.

**Funkcjonalność**

* realizacja projektu za pomocą metodyki Scrum,
* rozdzielanie zadań między pracowników,
* definiowanie produktu za pomocą user stories,
* podział użytkowników na role,
* zarządzanie Product Backlog,
* zarządzanie zgłoszeniami, szacowanym czasem, przepracowanym czasem,
* zarządzanie sprintami,
* tablica Kanban,
* zgłaszanie błędów oprogramowania, możliwość opisania powodów ich powstawania oraz załączenia plików, m.in. zrzutów ekranu,
* generowanie raportów - wykresów burndown, raport aktywności,
* łączenie commitów z programu GIT z zadaniami.

**Wymagania niefunkcjonalne**

* otwarty kod źródłowy, repozytorium na github.com
* prosta instalacja i uruchomienie, wraz z opcją instalacji w trybie lokalnym,
* responsywny web design, czyli zaimplementowanie automatycznego dostosowania wyglądu aplikacji do rozmiaru ekranu na którym jest wyświetlana (monitora, tabletu lub telefonu).
* intuicyjna obsługa narzędzia,
* modularna implementacja, otwarta na eksperymentowanie z rozszerzeniami metodyki Scrum.

### 2. Analiza użytkownika

Realizowany przez nas projekt skierowany jest głównie do trzech grup użytkowników:

* studentów informatyki,
* pracowników małych firm informatycznych,
* społeczności startupowej.

#### Analiza ograniczeń

W przypadku tego projektu nie uwzględniamy daleko idących ograniczeń użytkownika. Jedynymi ograniczeniami wartymi uwzględnienia są te wynikające z konktekstu użytkownika - konieczność dzielenia uwagi, ewentualne rozproszenie użytkownika.

#### Utworzenie person

##### Jan Niezbędny



Wiek: 29

Wykonywany zawód: programista Java projektu startupowego

Miejsce zamieszkania: Gdynia

Rodzina: narzeczona, rodzice i dwie siostry mieszkają w Krakowie

Hobby: układanie kostki Rubika na czas, serial “Gra o Tron”

Motto: “Każdy musi samemu zadbać o swój sukces.”

Umiejętności techniczne: dobra umiejętność posługiwania się komputerem,

znajomośći nowych technologii

znajomość języka Java

znajomość metodyk zwinnych

Jan jest programistą, niedawno skończył studia na kierunku Informatyka na Uniwersytecie Gdańskim. Zdecydowanie jest człowiekiem towarzyskim, typem “luzaka”. Jego marzeniem jest stworzyć własną firmę informatyczną. Cechuje go fakt bycia nocnym markiem, lubi pracować późnymi wieczorami.

Jana czeka niełatwe zadanie. Aplikacja ITDocHelper to pierwszy profesjonalny projekt, który współtworzy. Wierzy on, że jest to jego klucz do świata biznesu. Jest to aplikacja tworzona wraz z kolegami na zlecenie inwestora i ma ona na celu ułatwienie dokumentowania projektów IT. Inwestor nie opłaca kosztów narzędzi, jedynie koszty wytwarzania, serwerów i marketingu aplikacji.

Do zrealizowania celu, Jan potrzebuje narzędzia, która pomoże mu porządkować zadania i ułatwić zgłaszanie błędów przez testerów aplikacji. Zgłaszanie to musi być bardzo proste, ponieważ głównymi testerami Jana są jego koledzy, którzy nie znają się dobrze na nowych technologiach.

Doświadczenia:

* Jan posiada wiedzę na temat metodyki Scrum, miał na studiach przedmiot poświęcony metodykom Agile,
* Jan tworzył wraz z kolegami kilka projektów na studiach, wspomagając się tablicą Kanban, używali do tego tablicy korkowej i kolorowych karteczek.

Nawyki:

* Ze względu na pracę motodą Kanban, Jan przywykł do używania tablicy korkowej i karteczek.
* Jan jest typem nocnego marka, lubi pracować późnymi wieczorami.

Cele:

* Jan oczekuje, że aplikacja da mu możliwość korzystania z tablicy Kanban, ponieważ jest to rozwiązanie, do którego jest przyzwyczajony,
* Jan oczekuje, że aplikacja da możliwość łatwego zgłaszania błędów,
* Jan oczekuje, iż rozwiązanie będzie darmowe, ze względu na brak środków pieniężnych na narzędzia wspomagajace proces wytwarzania,
* Jan oczekuje, iż narzędzie ułatwi komunikację wewnątrz zespołu, zwłaszcza, że pozostali członkonkowie będą pracowali w innych godzinach niż sam Jan.

##### 

##### 

##### Zuzanna Pluta



Wiek: 21

Wykonywany zawód: studentka informatyki na Politechnice Gdańskiej

Miejsce zamieszkania: Gdańsk

Rodzina: mieszka wraz z rodziną w Gdańsku na Orunii Górnej

Hobby: makijaż i podróże, oglądanie seriali

Motto: “Zalety należy eksploatować, a wady minimalizować.”

Umiejętności techniczne: dobra umiejętność posługiwania się komputerem,

znajomość języka C#,

znajomość mtodyk zwinnych

Zuzanna jest studentką czwartego semestru kierunku Informatyka na Politechnice Gdańskiej. Marzy o pracy w dużej korporacji, takiej jak Microsoft czy Google. Jest ona typem introwertyka, oczekuje, że w firmie będzie miała własny pokój biurowu z widokiem na morze lub góry. Zuzanna zdecydowała się na kierunek Informatyka za namową ojca, który pracuje w branży IT od ponad 30 lat.

Na studiach kilkukrotnie miała doczynienia z metodyką Scrum, gdyż jest to preferowany przez prowadzących model prowadzenia projektów. W projekcie często pełni funkcję scrum mastera. Zawsze współpracuje z tą samą grupą znajomych.

Doświadczenia:

* Zuzanna pracowała w wielu projektach na studiach prowadzonych metodyką Scrum,
* Często pełni funkcję Scrum Mastera,
* Bardzo ceni codzienne scrumowe spotkania,
* Często korzysta z narzędzia Acunote.

Nawyki:

* Zuzanna lubi uporządkowany i ustrukturalizowany tryb pracy,
* Zuzanna lubi kończyć projekty na uczelnie przed wyznaczonym czasem, zabiera się za nie z wyprzedzeniem.

Cele:

* Zuzanna liczy, że narzędzie uporządkuje sposób pracy jej i zespołu,
* Ma nadzieję, że pozwoli na sprawieliwy podział pracy.

##### Jacek Tracz



Wiek: 35

Wykonywany zawód: analityk biznesowy, product owner w małej firmie

Miejsce zamieszkania: Elbląg

Rodzina: mieszka razem z małżonką, brat mieszka w Warszawie,

a rodzice w Elblągu

Hobby: jazda na Quadach oraz survival

Motto: “Czas to pieniądz.”

Umiejętności techniczne: dobra znajomość systemów operacyjnych z rodziny

Windows i Unix, dobra znajomość bazy danych Oracle,

ponad 5 lat doświadczenia w roli product ownera

Jacek jest analitykiem i CEO w małej firmie o nazwie MyITSolutions. Marzy o tym, aby jego firma rozrosła się do firmy wieloodziałowej, a nawet międzynarodowej. Jacek jest ambitny i bezwzględny, ceni sobie dobrze wykonaną pracę.

Jacek lubi pracować na wielu systemach operacyjnych, bardzo popiera ruch Open-Source, płaci dotacje na niektóre projekty. Pracowal już z użyciem wielu narzędzi wspomagających metodykę Scrum, jednak rozwiązania komercyjne często nie odpowiadają jemu potrzebom ze względu na niską modyfikowalność i konfigurowalność.

Doświadczenia

* Ponad 5 lat doświadczenia z narzędziami taki jak Jira, Acunote oraz IceScrum

Nawyki:

* Jacek lubi pracować na wielu systemach operacyjnych.

Cele:

* Jacek liczy na to, że narzędzie będzie umożliwiało pracę na wielu systemach operacyjnych,
* Jacek oczekuje, iż narzędzie będzie zawierało otwarty kod źródłowy, w celu możliwości modyfikacji na potrzeby firmy,
* Jacek szuka narzędzia wysokiej jakości,
* Ceni sobie możliwość personalizowania narzędzi.

### 3. Badanie użyteczności

#### 3.1. Przedmiot badania

Ponieważ rozwijana aplikacja nie jest jeszcze w fazie pozwalającej na jej badanie, a na rynku istnieje duża konkurencja, zbadane zostanią inne aplikacje realizujące podobne funkcjonalności. Analiza pomoże nam zidentyfikować funkcjonalności, na których zależy użytkownikom oraz określi wyróżnik projektu.

Ponieważ na rynku istnieje bardzo dużo rozwiązań tego typu, niemożliwe jest wybranie jednej konkretnej aplikacji do zbadania. Skupimy się na tym jak użytkownicy postrzegają wykorzystywane przez siebie programy oraz jakie funkcjonalności uznają za przydatne.

Niektóre dostępne na rynku aplikacje i serwisy Scrumowe:

* Acunote
* Jira
* Microsoft Tram Foundation Server
* Taiga
* IceScrum
* XPlanner
* Redmine
* EMForge
* BananaScrum

Opis i funkcjonalności oferowane przez aplikacje Scrumowe, na przykładzie Acunote:

Acunote to aplikacja do zarządzania projektami i oprogramowaniem wytwarzanym w metodyce Scrum zarówno dla małych zespołów jak i dużych firm. Umożliwia współpracę i komunikację wszystkich osób w firmie. Posiada duże możliwości analizy - przedstawianie postępów na wykresach typu burndown, tworzenie raportów. Jego dodatkową funkcją jest integracja z narzędziami do śledzenia błędów i z systemami kontroli wersji.

Funkcjonalności Acunote:

* tworzenie projektów podzielonych na różne zespoły / działy
* planowanie sprintów
* dodawanie zadań, edytowanie, usuwanie
* tworzenie backlogów
* wyszukiwanie zadań
* wykresy spalenia
* analizy i raporty
* tworzenie wiki
* kontrola dostępu dla użytkowników
* integracja z systemami kontroli wersji
* integracja z narzędziami do śledzenia błędów
* skróty klawiszowe

#### 3.2. Plan badania

Wybrane metody przeprowadzenia badań:

1. Wywiad
2. Ankiety online
3. Obserwacyjne z uzyciem scenariusza

**Wywiad:**

Wywiad zostanie przeprowadzony z programistą, który pracuje z wykorzystaniem metodyki Scrum i używa tego typu narzędzi na co dzień.

Zdecydowaliśmy się na taką formę badania, ponieważ rozmowa twarzą w twarz daje zazwyczaj dużo więcej szczegółowych informacji niż ankiety. Zależy nam na jakości uzyskanych danych, a nie na ich ilości.

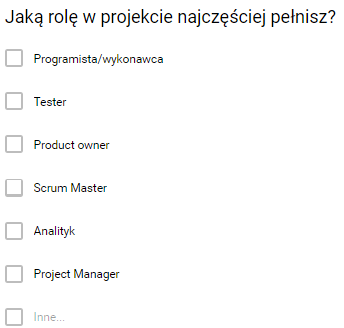
Planowany zakres zagadnień do poruszenia w wywiadzie:

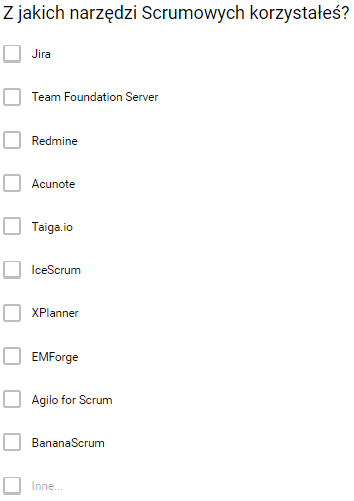
* Działanie procesu scrumowego:
  + Opisz w jaki sposób u was w firmie realizowany jest proces scrumowy.
* Używane narzędzia:
  + Jakiego narzędzia używacie do zarządzania procesem scrumowym?
  + Czy prosto było nauczyć się stosowanego narzędzia?
  + Czy nowe osoby szybko uczą się narzędzia? Co sprawia im problemy?
  + Czy interfejs narzędzia pozwala Ci szybko wykonywać twoją pracę? Czy coś jest źle rozwiązane w interfejsie?
  + Jakie funkcjonalności są według Ciebie dobrze zrealizowane/podobają Ci się?
  + Co jest kiepsko rozwiązane lub czego brakuje w obecnie stosowanym narzędziu?
* Priorytety funkcjonalności:
  + burndown chart’y,
  + tablica Kanban z taskami,
  + mierzenie czasu poświęconego na wykonywanie danego tasku,
  + automatyczna estymacja czasu ukończenia projektu na bazie user points i poprzednich projektów,
  + możliwość zgłaszania isssues,
  + integracja zadań i zgłoszeń z systemem wersji.
* Oczekiwania pozafunkcjonalne:
  + Czy byłbyś zainteresowany tworzeniem własnych wtyczek do używanego narzędzia w celu jak najdokładniejszego dostosowania go do realiów panujacych w miejscu Twojej pracy?
  + Jak ważna jest dla Ciebie możliwość pracy z wykorzystaniem danego narzędzia wspomagającego procesy projektowe w trybie lokalnym?
  + Często korzystasz w pracy z urzadzeń mobilnych i/lub posiadających niestandardowe rozmiary wyświetlacza (względem tradycyjnych ekranów komputerowych)?

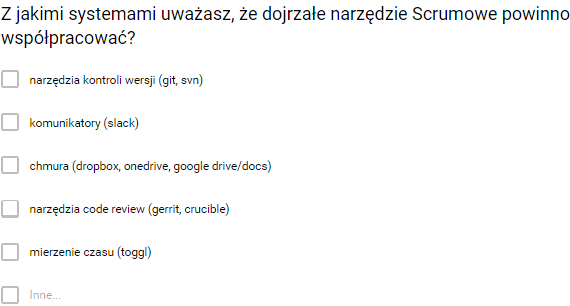
**Ankieta dla studentów i programistów:**

Ankiety online zostaną skierowane do studentów ze względu na łatwą możliwość dotarcia do dużej liczby osób. Planujemy zebrać ponad 50 odpowiedzi. Studenci zazwyczaj mają mniejsze doświadczenie z rozwiązaniami zwinnymi, więc w ankietach planujemy poznać ich oczekiwania i pierwsze wrażenia z użytkowania. Mamy nadzieję, że zebrane wyniki pomogą utworzyć nam rozwiązanie, które będzie maksymalnie przystępne dla nowych użytkowników.

W części pytań zmodyfikowaliśmy skalę Likerta usuwając opcję środkową w celu uniknięcia odpowiedzi, które nic by nie wniosły.



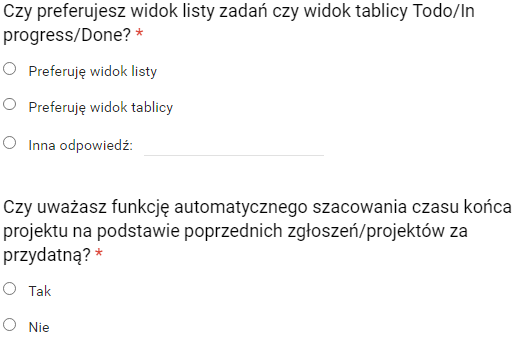


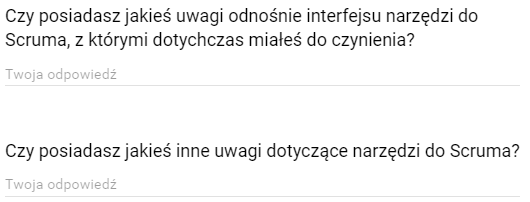












**Badanie metodą listy zadań do wykonania**

Badanie zostanie przeprowadzone na ochotnikach, studentach. Osoba badana dostanie do wykonania listę zadań w nieznanym do tej pory dla niej narzędziu. Całość pracy ma zostać nagrana.

Wybór programu padł na Acunote, gdyż zajął on wysoką pozycję w badaniu używanych przez studentów narzędzi i do tego jest łatwo dostępne dla każdego.

W badaniu chcielibyśmy sprawdzić jak nowy użykownik jest w stanie poradzić sobie z rozwiązaniami zastosowanymi w popularnym narzędziu. Stanowiłoby to dla nas cenną informację o dobrych/złych metodach realizacji funkcjonalności.

Użyta lista zadań do wykonania:

* Stworzyć Product Backlog z trzema pozycjami: “A”, “B”, “C” o priorytetach kolejno: P1, P2, P3;
* Utworzyć task “AAAA” i dodać mu podzadanie “BBBB” o priorytecie P2 , “CCCC” o priorytecie P3 oraz “DDDD” o prorytecie P4;
* Utworzyć sprint trwający dwa tygodnie i przypisać mu zadanie “AAAA”, “BBBB” i “CCCC”;
* Przypisać do siebie zadanie “BBBB” i wyestymować je na 4 Story Pointy
* Zakończyć task o nazwie “BBBB”;
* Ustawić task “CCCC” na częsciowo zrobiony;
* Do taska “DDDD” dodać komentarz: “Nie mamy czasu na wykonanie tego zadania, trzeba z niego zrezygnować.”;
* Zmienić nazwę taska “AAAA” na “AA”;

#### 3.3. Realizacja badania

Link do ankiety utworzonej za pomocą Google Forms został zamieszczony na forum studentów informatyki, którzy stanowią ważną grupę docelową projektu i często pracują już zawodowo. Udało się zdobyć 49 odpowiedzi w tym głównie od programistów, ale także project managerów, analityków biznesowych, srum masterów oraz product ownerów, dzięki czemu pokryty został niemal cały obszar grupy docelowej.

Przeprowadzony został jeden wywiad z pracownikiem firmy informatycznej. Zdecydowano się wybrać reprezentanta najważniejszej i najszerszej dla nas grupy przyszłych użytkowników - programistów. Zależało nam na przeprowadzeniu wywiadu z doświadczonym programistą, który miałby wieloletnie doświadczenie w użytkowaniu narzędzi do zarządzania projektami Scrum. Ze względu na niewielki dostęp do tej grupy, postaraliśmy się znaleźć pojedynczego odpowiedniego reprezentanta, który spełniałby zasadę CRACK, czyli byłby chętny do współpracy (Collaborative), był reprezentatywny (Representative), upoważniony (Authorized), zaangażowany (Committed) oraz posiadający szeroką wiedzę z badanej dziedziny (Knowledgable).

Na badanie metodą listy zadań do wykonania zgodziła się dwójka studentów, która nie miała wcześniej styczności z narzędziem Acunote. Ochotnicy zgodzili się na nagrywanie twarzy podczas wykonywania zadanych czynności. Na start badania osoby obserwowane otrzymały listę zadań do wykonania oraz dane do konta na Acunote. Nagrano obraz z ekranu oraz twarz i komentarze głosowe podczas wykonywania zadań. W czasie przeprowadzania badań, zidentyfikowano potrzebę przeprowadzenia rozmowy retrospekcyjnej po zakończeniu wykonywania zadań w celu lepszego poznania odczuć obserwowanego względem narzedzia.

#### 3.4. Wyniki badania

**Wywiad**

Wywiad został przeprowadzony z Senior Developerem w jednej z gdańskich firm informatycznych.

*Działane procesu scrumowego*

Opisany przez programistę proces scrumowy był lekko zmodyfikowany w stosunku do standardowego. Przede wszystkim sprinty w projekcie nie są jednakowej długości - mają od 5 do 10 dni. Konsekwencją tego jest brak ustalonych dni przeprowadzania planowań sprintów, retrospektyw, itp. Nie są przeprowadzane Sprint Review ze względu na to, że klient ma ciągły dostęp do produktu i kodu. Estymowanie zadań wykonywane jest od razu w godzinach - nie używa się Story Pointów. Z tego powodu około 40% sprintów nie jest zamykanych. Niedokończone zadania nie przechodzą z powrotem do Product Backloga, lecz sprint jest przedłużany aż wszystko zostanie wykonane.

*Używane narzędzia*

Obecnie w projekcie używany jest **Team Foundation Server**. Narzędzie jest bardzo wygodne w użytkowaniu oraz łatwo przyswajalne ze względu na możliwość podpięcia całego interfejsu pod środowisko programistyczne Visual Studio, który jest nieustannie używany przez wszystkich programistów w projekcie.

Programista uważa TFS za narzędzie niemal idealne, ale zauważył kilka wad, które przeszkadzają nie tylko jemu, ale też wszystkim innym użytkownikom. Po pierwsze TFS nie ma “sensownego” złączenia gałęzi kodu w systemie kontroli wersji, przez co często programiści muszą zmagać się z częstym zjawiskiem zduplikowania kodu. Po drugie programista zauważył, że trudność sprawia mu to, że pod jednym skrótem klawiszowym może być podpiętych kilka akcji, przez co często zdarza mu się wykonać niezamierzone akcje niechcący.

Najczęściej wykorzystywaną funkcją w TFS przez niego jest repozytorium kodu. Bardzo podoba mu się fakt, że może ustawić warunki, które muszą być spełnione, by mógł udać się commit kodu (np. musi być pozytywna ocena z code review). Za najlepszą cechę TFS wskazuje to, że wszystko ma w jednym miejscu, bo do wszystkich funkcji ma dostęp z Visual Studio.

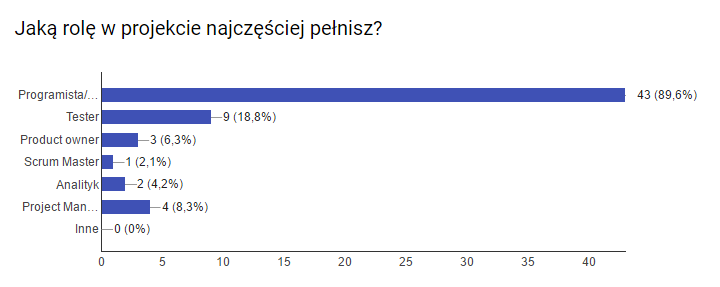
*Priorytety funkcjonalności*

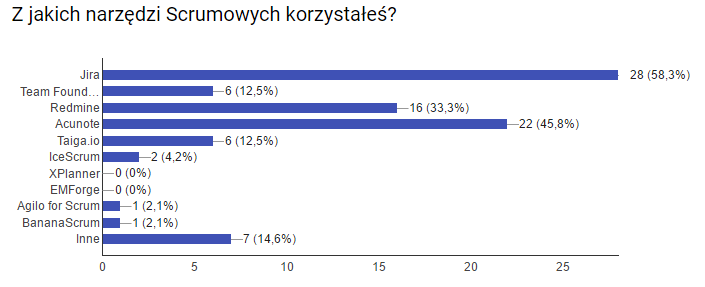
Najważniejsze dla programisty jest to, by narzędzie miało wbudowaną tablicę Kanban, posiadało integrację z kontrolą wersji oraz dawało możliwość dodawania nowych zadań. Za najmniej ważną fukcję uważa przypisywania godzin spędzonych nad zadaniem.

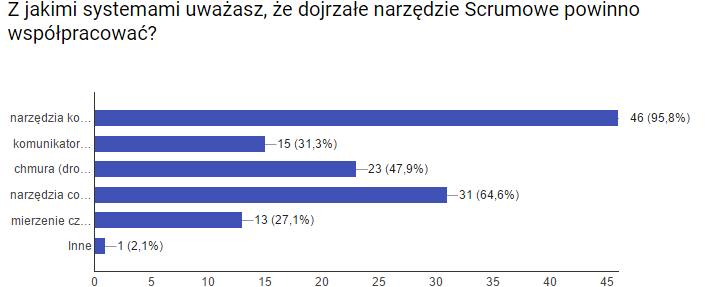
*Oczekiwania pozafunkcjonalne*

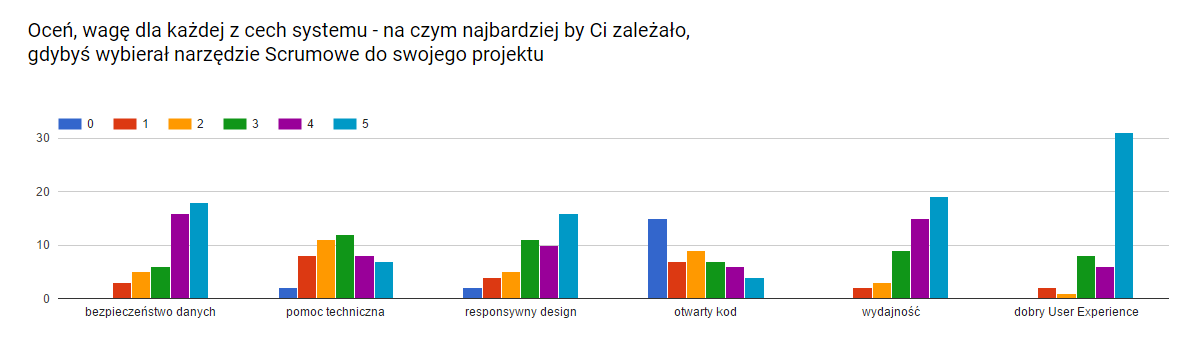
Programista wyraźnie zaznaczył, że nowe narzędzie zastosowałby w swoim projekcie tylko pod waunkiem, gdyby zapewniona była pomoc techniczna i istniało społeczeństwo użytkowników. Najbardziej mu zależy na bezpieczeństwie narzędzia.

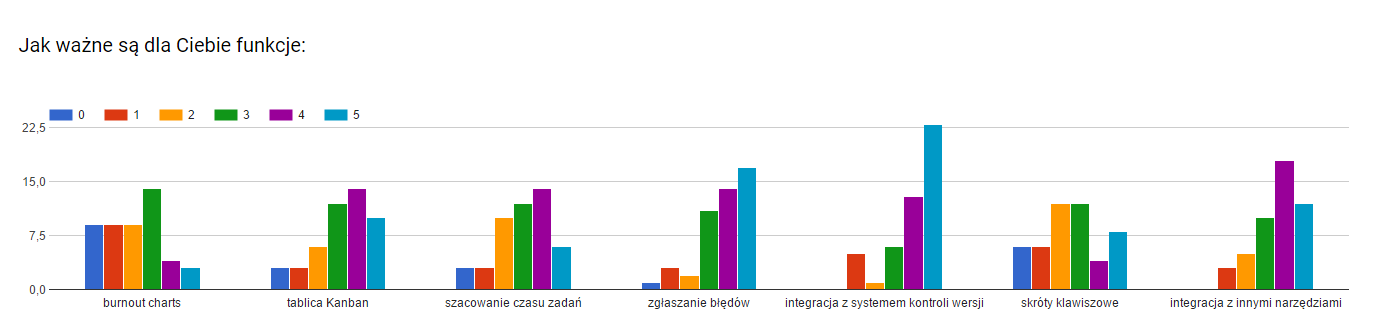
**Ankiety**

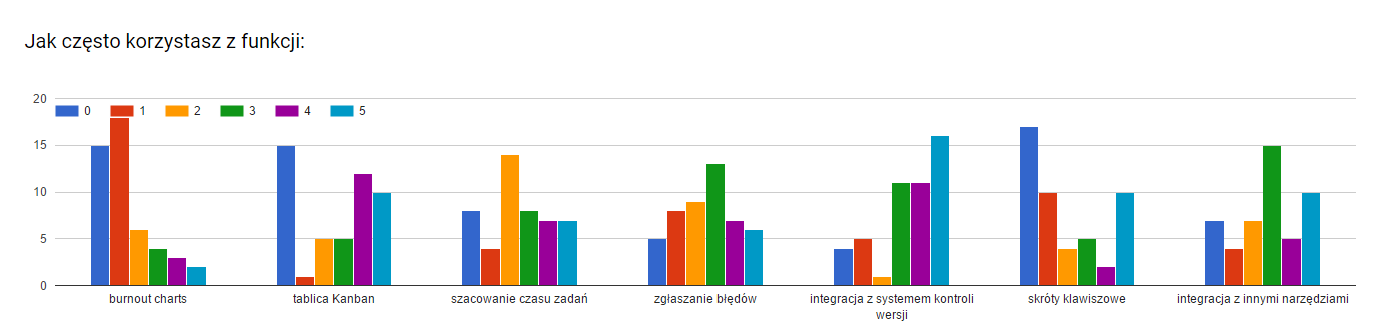


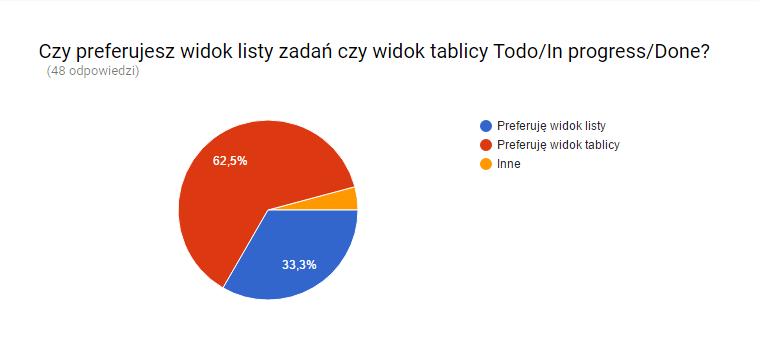


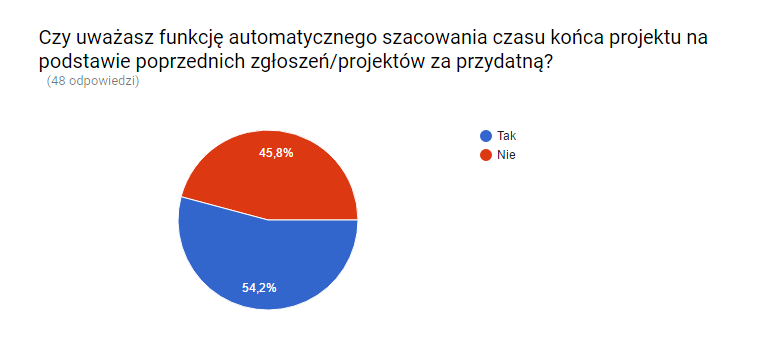


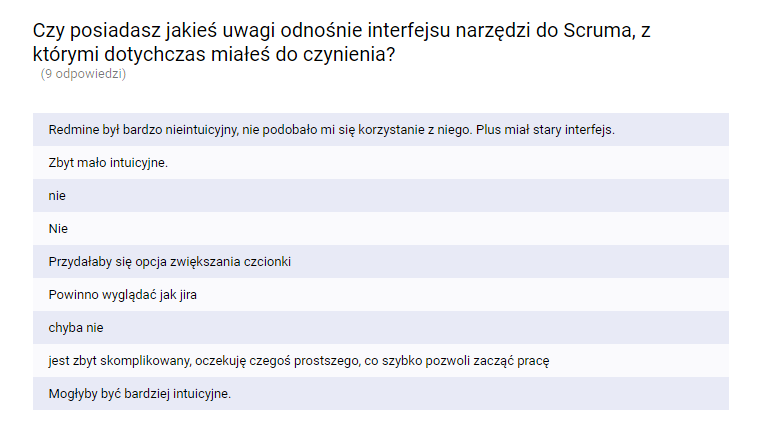


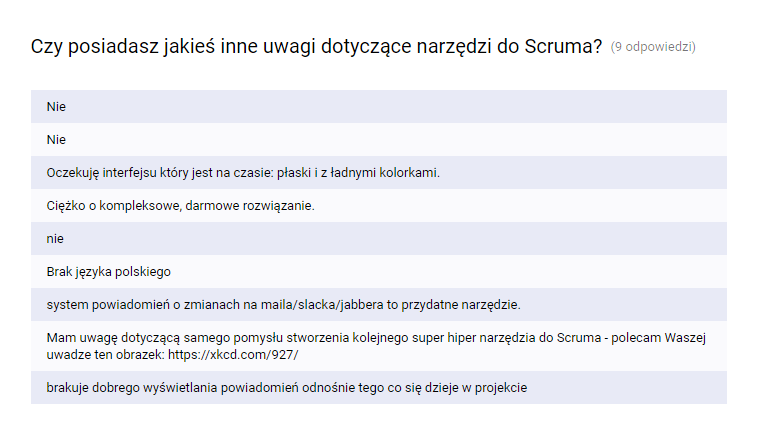












**Badanie metodą listy zadań do wykonania**

Otrzymano nagrania przedstawiające nowych użytkowników wykonujących listę zadanych czynności. Są one udostępnione na Google Drive: <https://drive.google.com/folderview?id=0B-60dLuXCbeYMW9QN2M4X0JFb0E&usp=sharing>

Badanie 1

Przeprowadzone badanie trwało około 10 minut. Badany był obserwowany przez jednego członka z zespołu, po czym przeprowadzono nie planowaną wcześniej rozmowę odnośnie wrażeń z aplikacji.

Badanie 2

Drugie badanie przeprowadzono zdalnie, nagrany został film je dokumentujący. Użytkownik został poproszony o komentowanie wszystkiego co robi.

#### 

#### 

#### 4. Wnioski z badania

**Wywiad**

Z wywiadu można wywnioskować, że narzędzia do zarządzania procesami scrum powinny być przygotowane na różne modyfikacje procesu - powinny być konfigurowalne pod względem długości sprintów, jednostki szacowania zadań, itp.

Narzędzie, z którymi obecnie respondent pracuje (TFS) jest bardzo chwalone pod względem przyjemności w użytkowaniu oraz łatwości nauki. Jego wady są głównie związane z błędnym działaniem funkcjonalności.

Najważniejszym aspektem dla programisty jest możliwość korzystania z tablicy Kanban, integracja narzędzia z kontrolą wersji oraz możliwość dodawania nowych zadań. Z kolei za najważniejsze cechy pozafunkcjonalne wskazane zostały bezpieczeństwo oraz wsparcie techniczne.

**Ankiety online**

Zdecydowanie najczęściej w ankiecie mieliśmy do czynienia z programistami. Uważamy, że to dobrze, gdyż wedle założeń Agile w zespołach nie ma ściśle określonych ról i każdy jest deweloperem, i co za tym idzie bierze udział w całym procesie wytwarzania oprogramowania, a nie tylko w jego wycinku.

Najczęściej używanymi narzędziami scumowymi okazały się Jira i Acunote. Postanowiliśmy w dalszych badaniach skupić się na tym drugim.

Respondenci są zgodni co do tego, że must-have dla takiego narzędzia jest możliwość współpracy z systemem kontroli wersji typu GIT czy SVN, co zostało również potwierdzone w odpowiedziach do kolejnych pytań.

*Na jakich cechach użytkownikom zależy najbardziej?*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nazwa cechy** | **bezpieczeństwo danych** | **pomoc techniczna** | **responsywny design** | **otwarty kod** | **wydajność** | **dobry User Experience** |
| **Uśredniony wynik** | 3,8 | 2,8 | 3,5 | 1,9 | 3,9 | 4,3 |

Obliczając uśredniony wynik dla każdej z zaprezentowanych w ankiecie cech, widzimy, że osobom pracującym w projektach IT najbardziej zależy na dobym User Experience. Ta potrzeba może świadczyć o tym, że odczuwają braki na tej płaszczyźnie w obecnie stosowanych narzędziach. Jedynie pomoc techniczna i otwartość kodu uzyskała wynik wskazujący na to, że nie są to specjalnie potrzebne cechy.

Ciekawym spostrzeżeniem jest sprzeczność opinii na temat pomocy technicznej biorąc pod uwagę ankietę i wywiad. Sądzimy, że wpłynął na to fakt, iż ankietowanymi byli studenci nie posiadający większego doświadczenia w branży, natomiast wywiad został przeprowadzony z osobą z wieloletnim doświadczeniem, czyli bardziej świadomej faktycznych potrzeb uczestników projektu informatycznego.

*Jakie funkcje ankietowani uważają za ważne?*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nazwa funkcji** | **burndown charts** | **tablica Kanban** | **szacowanie czasu zadań** | **zgłaszanie błędów** | **integracja z systemem kontroli wersji** | **skróty klawiszowe** | **integracja z innymi narzędziami** |
| **Uśredniony wynik** | 2,1 | 3,3 | 3,0 | 3,8 | 4,0 | 2,6 | 3,7 |

Tak, jak zostało zakomunikowane wcześniej (oczywiste funkcjonalności takie jak tworzenie product backlog, sprintów nie były brane pod uwagę), jako najważniejszą funkcję narzędzia srumowego ankietoIntwani wskazali integrację z systemem kontroli wersji. Jedynie skróty klawiszowe, które jak sądzimy używane są przez bardzo doświadczonych użytkowników oraz wykresy burndown zostały ocenione jako mało ważne.

*Z których funkcji ankietowani najczęściej korzystają?*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nazwa funkcji** | **burndown charts** | **tablica Kanban** | **szacowanie czasu zadań** | **zgłaszanie błędów** | **integracja z systemem kontroli wersji** | **skróty klawiszowe** | **integracja z innymi narzędziami** |
| **Uśredniony wynik** | 1,3 | 2,6 | 2,5 | 2,6 | 3,4 | 1,9 | 2,8 |

Widać wyraźną korelację pomiędzy częstością używania danej funkcji przez członka zespołu, a poczuciem jej wagi w całym procesie. Dla przypomnienia, by nie zmyliły nas niższe oceny, 2 oznaczała użycie mniej więcej raz na sprint, a 3 już kilka razy w tygodniu. Ta korelacja, oznacza że użytkownicy często mogą nie zdawać sobie sprawy z wagi danej funkcjonalności, jeżeli mało z niej korzystają. W przyszłości przy badaniu zadowolenia z korzystania z zaimplementowanych przez nas funkcjonalności chcemy rozróżnić użytkowników na dwie grupy - często i sporadycznie korzystających z danej funkcji.

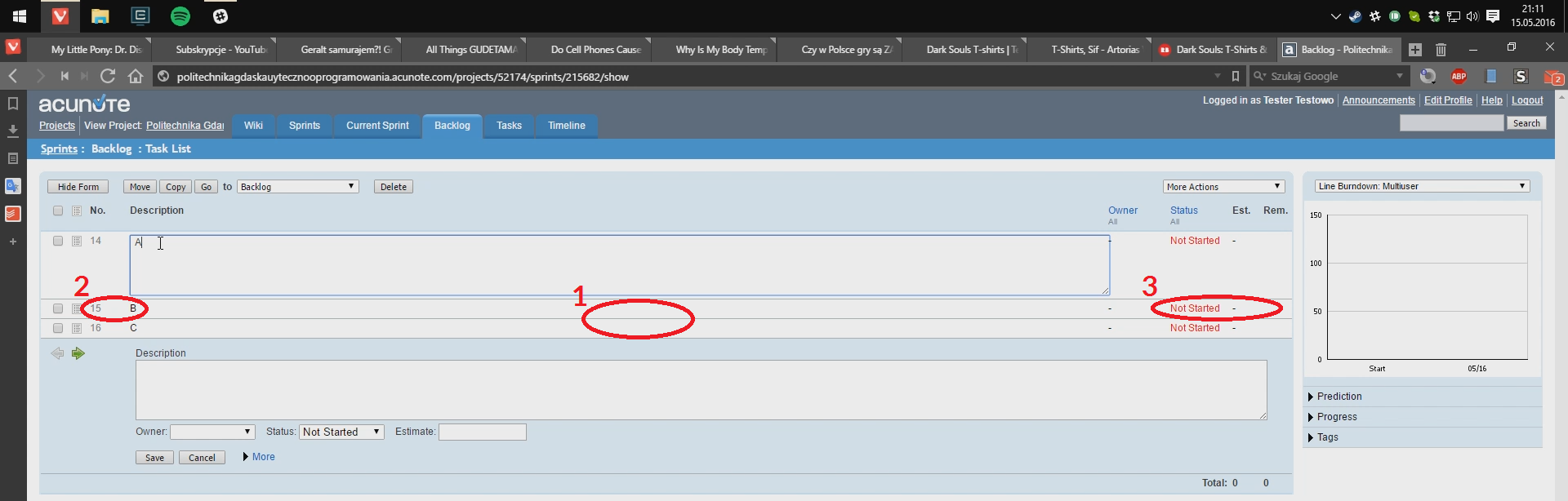
Powyższe tabele wyników oraz odpowiedzi na kolejne pytania z pewnością pomogą nam w priorytetyzacji oraz w kształcie wykonywanych zadań podczas wytwarzania narzędzia. Przykładowo dowiedzieliśmy się, że badani preferują widok tablicy to do/in progress/done zamiast listy zadań.

**Badanie metodą listy zadań do wykonania**

***Użytkownik 1***

*Informacje wstępne:* w momencie udostępnienia aplikacji użytkownikowi, okazało się, że bez włączenia funkcji ustawiania priorytetów dla tasków z poziomu administratora, wykonanie niektórych poleceń jest niemożliwe. Podczas tworzenia podsumowania wyszło na jaw, iż jeden z członków zespołu był świadom tego faktu. Nie poinformował jednak o tym reszty, by zobaczyć jak sytuacja problemowa zostanie rozwiązana.

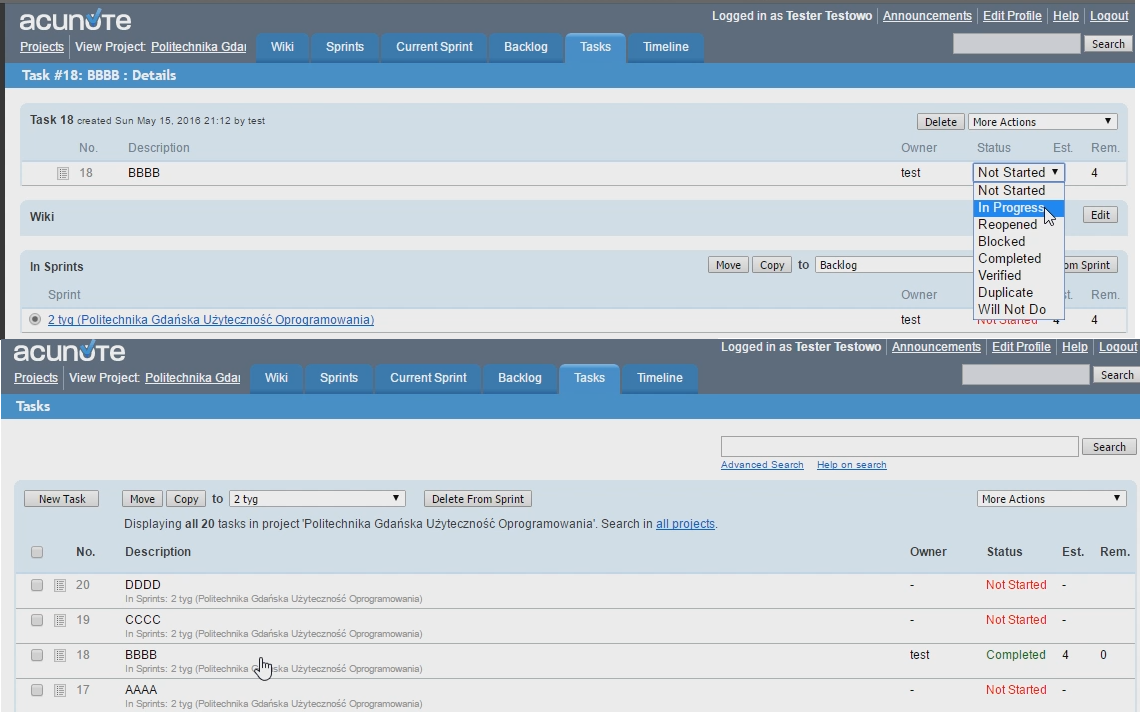
Po wykonaniu zadań przeprowadziliśmy rozmowę retrospekcyjną w celu identyfikacji elementów, które okazały się problematyczne dla użytkownika.



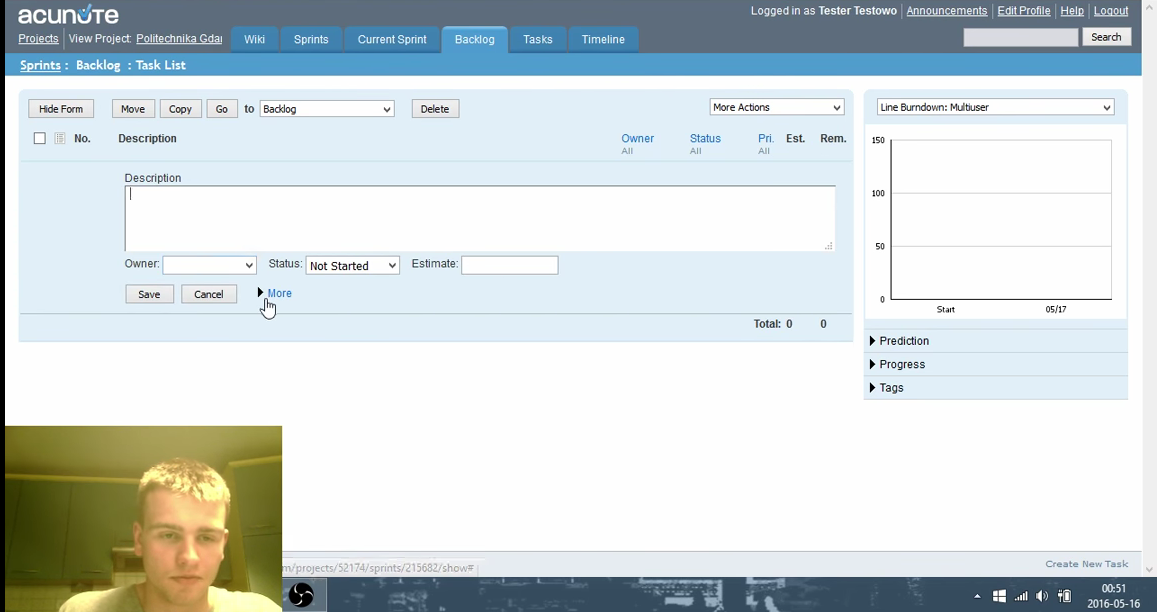
*Wnioski po obserwacjach i rozmowie:*

1. Użytkownikowi nie podobała się ilość białej przestrzeni jaka jest na środku ekranu, uważa że interfejs mógł zostać lepiej rozplanowany, by wykorzystać tę przestrzeń.
2. Nawet członek zespołu nie poradził sobie ze znalezieniem odpowiedniej opcji konfiguracyjnej dotyczacej priorytetów zadań, a co za tym idzie, użytkownik nie był w stanie wykonać niektórych poleceń
3. Kliknięcie w nazwę tasku nie przenosi do jego szczegółów, tylko pozwala zmienić jego nazwę. By przejść do szczegółów należy nacisnąć id tasku. Okazało się to mylące, użytkownik popełnił kilkukrotnie ten błąd.
4. Zadanie, które sprawiło użytkownikowi najwięcej problemów, to przypisanie do siebie taska. Użytkownik nie zorientował się, że elementy po prawej stronie są klikalnymi listami rozwijanymi. To zadanie zajęło ponad dwie minuty (4:20-6:40), wzbudziło ono najwięcej frustacji, którą można zaobserwować na filmie około 4:50.

Mimo iż użytkownikowi udało się wykonać to zadanie, to wprowadzał zmiany w szczegółach taska, mimo iż ta sama funkcjonalość była dostępna poziom wyżej. (zdj. poniżej).

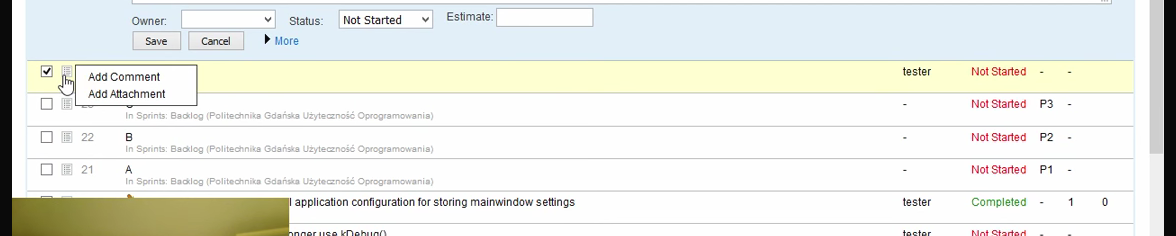


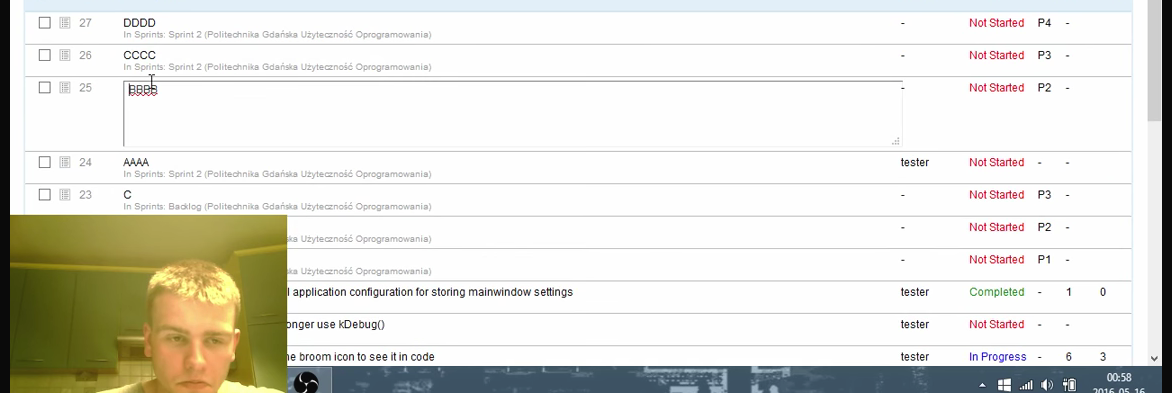
***Użytkownik 2***



*Wyniki po obserwacjach:*

1. Użytkownik zastanawiał się przez około 10 sekund, gdzie wpisać nazwę zadania w Product Backlog. Pole *Description* częściej sie ludziom kojarzy z dodatkowym opisem niż z tytułem/tematem zadania.
2. Użytkownik długo zastanawiał się, w jaki sposób stworzyć podzadanie. W ostateczności udało mu się dodać realację do zadania, ale nie utworzył odpowiednich podzadań. Tworząc relację próbował wpisać nazwę zadania, ale wyskoczył błąd, gdyż program tworzy relacje po identyfikatorze zadania, a nie po jego nazwie. Użytkownik musiał się cofnąć, zapamiętać identyfikatory zadań i wpisać ręcznie ich numery.
3. Przenosząc task do Sprintu 2 użytkownik szukał takiej opcji w menu rozwijanym dla zadania, jednak ujrzał tylko opcje *Add comment* i *Add attachment*.
4. Wielokrotnie użytkownik próbował przejść do szczegółów zadania klikając na jego nazwę, wtedy zamiast strony ze szczegółami pojawiło się pole do edycji. Dopiero po kliknięciu na identyfikator zadania użytkownik został przeniesiony we właściwe miejsce.





**Wnioski z obserwacji**

Dzięki obserwacjom udało się zidentyfikować elementy interfejsu, na które trzeba będzie zwrócić szczególną uwagę podczas wytwarzania aplikacji. W obu badaniach użytkownicy popełnili ten sam błąd: przy przechodzeniu do szczegółów aplikacji klikając na jej nazwę zamiast na identyfikator. Jest to rozwiązanie, którego należy uniknąć. Po zaprojektowaniu interfejsu chcemy sprawdzić, czy użytkownicy poprawnie zidentyfikują wszystkie klikalne elementy. Adresuje to sytuację problemową użytkownika 1, gdzie przez długi czas nie zorientował się, że element jest klikalny. Ważną informacją dla nas jest, że korzystanie z identyfikatorów zadań jest dla użytkowników nieintuicyjne i należy ich unikać na rzecz nazw zadań lub przeciągania interesujących elementów we właściwe miejsca.

**Podsumowanie**

Podsumowując wszystkie zebrane wyniki, bezkonkurencyjną cechą dobrego narzędzia scrumowego jest według respondentów dobry user experience, a więc wysoka jakość możliwych interakcji z systemem. Szczególnie widać to w wynikach obserwacji i odpowiedziach na pytania otwarte ankiety. Jest to więc rzecz, do której przywiążemy szczególną uwagę przy projektowaniu własnego rozwiązania. Interesujące dla nas było to, że interfejs aplikacji Acunote został określony przez użytkowników podczas rozmów po obserwacji jako przestarzały, ale jednocześnie aplikacja ta wypadła na drugim miejscu w ankietach pod względem liczby osób mających z nią styczność. Bardziej popularna okazała się tylko Jira.

Funkcją oprogramowania, która jako ważna, pojawiała się najczęściej zarówno w ankietach, jak i w wywiadzie z osobą posiadającą duże doświadczenie w branży IT, jest możliwość integracji z systemem kontroli wersji typu GIT lub SVN. Poważnym błędem byłoby zignorować tak znaczącą informację. Dlatego też jest to dla nas najważniejsza funkcjonalność poza tymi ściśle związanymi z przeprowadzeniem procesu scrumowego. Kolejnym priorytetem jest możliwość zgłaszania błędów.

Wydajność oraz bezpieczeństwo są to cechy pozafunkcjonalne, które respondenci określili jako szczególnie dla nich ważne. Ta informacja wnosi bardzo dużo na poziomie samego projektu systemu. Dzięki niej będziemy mieli szansę uniknąć błędów w zalożeniach, które później mogłyby okazać się zbyt kosztowne do naprawienia.

Mimo, że funkcja tworzenia tablicy Kanban została oceniona nieco gorzej w ankietach, w otoczeniu innych funkcjonalności, zdecydowaliśmy się na jej implementację. Głównym powodem było przypisanie jej dużej wagi podczas przeprowadzonego wywiadu, jak i zdanie większości ankietowanych o wyższości takiego widoku nad zwykła listą zadań.

Rzeczą, na którą również zwrócimy uwagę, jest kwestia dostępności pomocy technicznej. Mimo tego, że w ankietach została ona oceniona jako mało znacząca, reprezentant typu CRACK profesjonalnego środowiska programistów ocenił ją jako absolutny must-have dla wytwarzanego narzędzia. Postanowiliśmy zaufać jego wiedzy, a przede wszystkim doświadczeniu.

Ze zwględu na brak zainteresowania użytkowników, zdecydowaliśmy nie skupiać się na poniższych cechach/funkcjonalnościach:

* otwarty kod,
* skróty klawiszowe,
* wykresy burndown.

Natomiast responsywny design oraz szacowanie czasu zadań zamierzamy zaimplementować tylko w przypadku nadwyżki czasu i zasobów.