Menadżer zadań – MemoTime

Biel Patryk, Janusz Anna, Oleksy Kamil

Zespołowe przedsięwzięcie inżynierskie

Informatyka

Rok. akad. 2017/2018, sem. I

Prowadzący: dr hab. Marcin Mazur

Spis treści

1 Opis projektu

1.1 Członkowie zespołu

1. Anna Janusz (kierownik projektu)
2. Patryk Biel
3. Kamil Oleksy

1.2 Cel projektu (produkt)

Cel projektu to stworzenie aplikacji "MemoTime", dzięki której można efektywniej organizować swój czas.

1.3 Potencjalny odbiorca produktu (klient)

Potencjalnym klientem będzie każda osoba z dostępem do internetu, ponieważ nasza aplikacja będzie działała na serwerze internetowym. Grupą najbardziej potencjalnych odbiorców będą informatycy, osoby pracujące, które lubią planować czas, jak i osoby które chcą "uporządkować" swój czas.

1.4 Metodyka

Projekt będzie realizowany przy użyciu (zaadaptowanej do istniejących warunków) metodyki *Scrum*.

2 Wymagania użytkownika

Struktura historyjek (User Story):
Jako *kto/kiedy/gdzie*, **chcę** *co*, **ponieważ** *dlaczego* **warunki satysfakcji**.

2.1 User story 1

Jako użytkownik chcę mieć możliwość zarejestrowania się i zalogowania się do aplikacji za pomocą adresu e-mail lub loginu, oraz hasła, bo chce mieć zagwarantowane bezpieczeństwo i pewność że nikt nie będzie widzieć moich planów.

2.2 User story 2

Jako użytkownik chcę mieć możliwość dodawania projektów (zbiorów zadań), zadań do projektów bo pozwoli mi się to skupić na jednym zadaniu i lepiej zorganizować sobie czas.

2.3 User story 3

Jako użytkownik chcę mieć możliwość modyfikowania/ usuwania zadań bo być może zmienię zdanie bądź zadanie wygaśnie, a chce mieć możliwość ich edycji.

2.4 User story 4

Jako użytkownik chcę mieć możliwość zaznaczania zadań, które zostały wykonane, bo będę widział co już zostało zrobione i nie będę zawracał sobie już tym głowy.

2.5 User story 5

Jako użytkownik chcę mieć możliwość priorytetowania zadań / etykietowania zadań, bo pozwoli mi widzieć to co jest ważniejsze do zrobienia i pomoże mi szybciej szukać zadań, które mam do wykonania (np. kontekst - w domu).

2.6 User story 6

Jako użytkownik chcę mieć możliwość dodawania podzadań do zadania, dzięki czemu będę mógł tworzyć tzw. checklisty i robić listę zakupów czy listę rzeczy do zrobienia na studia.

2.7 User story 7

Jako użytkownik chcę mieć możliwość przypominania mi o zadaniach do wykonania przez wysyłanie dziennego zrzutu zadań na e-mail, bo czasem nie będę mieć dostępu do komputera a nie chce o niczym zapomnieć.

2.8 User story 8

Jako użytkownik chcę mieć możliwość drukowania zadań/projektów, aby np. powiesić je sobie na lodówce, bo będą mi przypominały o tym co jest do zrobienia.

2.9 User story 9

Jako użytkownik chcę mieć możliwość widoku kalendarza i dodatkowej funkcji dodawania specjalnych dat(np. urodziny) ze specjalną ikonką , aby wzrokowo widzieć ważne dla mnie daty jak i rozkład swoich zadań.

3 Harmonogram

3.1 Rejestr zadań (Product Backlog)

- Data rozpoczęcia: 31.10.2017
- Data zakończenia: 12.11.2017.

3.2 Sprint 1

- Data rozpoczęcia: 13.11.2017
- Data zakończenia: 21.11.2017
- Scrum Master: Patryk Biel
- Product Owner: Anna Janusz
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

3.3 Sprint 2

- Data rozpoczęcia: 21.11.2017«data».
- Data zakończenia: 05.12.2017«data».
- Scrum Master: Patryk Biel
- Product Owner: Anna Janusz
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

3.4 Sprint 3

- Data rozpoczęcia: 05.12.2017«data».
- Data zakończenia: 19.12.2017«data».
- Scrum Master: Patryk Biel
- Product Owner: Anna Janusz
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

3.5 Sprint 4

- Data rozpoczęcia: 19.12.2017«data».
- Data zakończenia: 02.01.2018«data».
- Scrum Master: Kamil Oleksy
- Product Owner: Anna Janusz
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

3.6 Sprint 5

- Data rozpoczęcia: 02.01.2018«data».
- Data zakończenia: 09.01.2018«data».
- Scrum Master: Anna Janusz
- Product Owner: Kamil Oleksy
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

3.7 Sprint 6

- Data rozpoczęcia: 09.01.2018«data».
- Data zakończenia: 16.01.2018«data».
- Scrum Master: Anna Janusz
- Product Owner: Kamil Oleksy
- Development Team: Kamil Oleksy, Patryk Biel, Anna Janusz

4 Product Backlog

4.1 Backlog Item 1

Tytuł zadania. Instalacja frameworka Angular.

Opis zadania. Pobranie środowiska node.js, następnie w oparciu o oficjalną dokumentację, za pomocą menadżera pakietów npm pobranie frameworka Angular, a następnie uruchomienie aplikacji testowej.

Definition of Done. Wchodząc na stronę na której uruchomiona jest aplikacja, po procesie instalacji frameworka Angular widoczna będzie strona powitalna potwierdzająca pomyślny proces instalacji.

4.2 Backlog Item 2

Tytuł zadania. Stworzenie strony startowej aplikacji WWW.

Opis zadania. Stworzenie szablonu HTML strony startowej wraz z odpowiadającymi jej stylami CSS, oraz komponentu menu głównego.

Definition of Done. Po wejściu na stronę główną aplikacji w centralnej części strony wyświetli się blok zawierający informacje o aplikacji. W górnej części strony dostępne będzie menu wraz z przyciskami prowadzącymi do podstron logowania oraz rejestracji.

4.3 Backlog Item 3

Tytuł zadania. Stworzenie interfejsu do rejestracji użytkownika.

Opis zadania. Stworzenie szablonu HTML zawierającego formularz rejestracyjny zawierający pola:

- e–mail
- nazwa użytkownika
- hasło
- data urodzenia
- akceptacja regulaminu korzystania z aplikacji

Definition of Done. Po wejściu na podstronę pojawia się formularz do rejestracji użytkownika

4.4 Backlog Item 4

Tytuł zadania. Stworzenie interfejsu do logowania użytkownika.

Opis zadania. Stworzenie szablonu HTML, który umożliwia wpisanie danych do zalogowania – login lub e–mail oraz hasło, a także przycisk funkcyjny zaloguj.

Definition of Done. Po wejściu na podstronę pojawia się formularz logowania użytkownika.

4.5 Backlog Item 5

Tytuł zadania. Widok z danymi użytkownika.

Opis zadania. Stworzenie widoku w środowisku Angular, który wyświetli dane zalogowanego użytkownika.

Definition of Done. Wchodząc w podstronę zalogowanego użytkownika zobaczymy jego dane (nick, avatar..).

4.6 Backlog Item 6

Tytuł zadania. Widok projektów oraz zadań.

Opis zadania. Stworzenie widoku w którym można będzie zobaczyć utworzone przez użytkownika zadania i projekty.

Definition of Done. Wchodząc na stronę zobaczymy strukturę naszych projektów oraz zadań.

4.7 Backlog Item 7

Tytuł zadania. Szablon email z zadaniami

Opis zadania. Stworzenie szablonu użytkownika, który będzie zawierał dane na temat zadań i projektów na każdy dzień.

Definition of Done. W podstronie gdzie znajduje się szablon e-maila zobaczymy zadania bądź projekty , które w danym dniu mają być przesłane do użytkownika na jego pocztę jako przypomnienie.

4.8 Backlog Item 8

Tytuł zadania. Interfejs do tworzenia projektu.

Opis zadania. Stworzony zostanie graficzny interfejs , gdzie użytkownik będzie mógł dodać projekt.

Definition of Done. Po wejściu na stronę pojawi się interfejs przy pomocy którego dodamy nowy projekt.

4.9 Backlog Item 9

Tytuł zadania. Interfejs do tworzenia nowego zadania.

Opis zadania. Stworzony zostanie graficzny interfejs, gdzie użytkownik będzie mógł dodać nowe zadanie.

Definition of Done. Wchodząc na stronę zobaczymy interfejs, gdzie użytkownik będzie miał możliwość stworzenia nowego zadania.

4.10 Backlog Item 10

Tytuł zadania. Zaprojektowanie interfejsu kalendarza.

Opis zadania. Stworzony zostanie interfejs kalendarza mający strukturę kafelek gdzie można będzie zobaczyć przy których dniach mamy zdanie/projekt.

Definition of Done. Wchodząc w podstronę zobaczymy kalendarz w formie kafelek , dzięki któremu zobaczymy nasze zadania/projekty z perspektywy całego miesiąca.

4.11 Backlog Item 11

Tytuł zadania. Widok pokazujący zakończone zadania.

Opis zadania. Stworzony zostanie widok, dzięki któremu zobaczymy projekty zakończone oraz te stworzone przez użytkownika.

Definition of Done. Po wejściu w podstronę zobaczymy listę projektów, które już zakończyliśmy oraz zadania które zostały stworzone.

4.12 Backlog Item 12

Tytuł zadania. Zaprojektowanie funkcjonalności etykietowania zadań.

Opis zadania. Stworzenie graficznych etykiet do zadań, które pomogą w lepszym zarządzaniu nimi.

Definition of Done. Przy każdym zadaniu można dodać etykietę lub ją usunąć.

4.13 Backlog Item 13

Tytuł zadania. Instalacja środowiska Docker wraz z silnikiem bazodanowym MSSQL.

Opis zadania. Instalacja środowiska Docker, a następnie pobranie oraz konfiguracja kontenera Docker zawierającego silnik bazodanowy MSSQL.

Priorytet. Normalny

Definition of Done. Pobrane oraz skonfigurowane środowisko Docker wraz z kontenerem zawierającym silnik bazodanowy MSSQL pozwala na utworzenie bazy danych, służącej jako magazyn na dane aplikacji, podłączenie się do niej oraz korzystanie z niej z poziomu aplikacji.

4.14 Backlog Item 14

Tytuł zadania. Stworzenie repozytorium użytkowników.

Opis zadania. Zaprogramowanie klasy repozytorium użytkowników.

Priorytet. Normalny

Definition of Done. W repozytorium zaimplementowane są metody do dodawania, usuwania, pobierania użytkownika oraz aktualizacji jego danych. Repozytorium komunikuje się z silnikiem bazodanowym w celu trwałego zapisu danych na dysku. Repozytorium może być wykorzystywane przez inne klasy np. serwisy służące do tworzenia nowego użytkownika.

4.15 Backlog Item 15

Tytuł zadania. Stworzenie repozytorium projektów.

Opis zadania. Zaprogramowanie klasy repozytorium projektów.

Priorytet. Normalny

Definition of Done. W aplikacji stworzona została klasa służąca jako repozytorium projektów. W klasie tej znajdują się metody do zapisywania nowo utworzonego projektu, pobierania go, modyfikowania oraz usuwania. Klasa repozytorium projektów komunikuje się z bazą danych w celu trwałego zapisu danych na dysku. Klasa ta może być wykorzystywana w innych klasach np. serwisów projektów. Klasa repozytorium projektów posiada swój własny interfejs, dzięki czemu inne klasy wykorzystujące klasę repozytorium, wiedzą jakie metody zaimplementowane są w klasie.

4.16 Backlog Item 16

Tytuł zadania. Stworzenie funkcjonalności rejestracji użytkowników.

Opis zadania. Zaprogramowanie klasy – serwisu służącej do rejestracji nowych użytkowników.

Priorytet. Normalny

Definition of Done. W aplikacji utworzona jest klasa UserService, wraz z metodą do rejestracji nowego użytkownika. Stworzona jest również klasa User odwzorowująca użytkownika aplikacji. Klasa ta posiada następujące pola:

- Identyfikator użytkownika,
- Nazwa użytkownika,
- E–mail użytkownika,
- Hasło użytkownika,
- Data rejestracji,
- Sól użyta do procesu haszowania hasła

W metodzie służącej do rejestracji, znajduje się walidacja adresu e–mail oraz nazwy użytkownika, która polega na sprawdzeniu czy przesłany przez użytkownika e–mail oraz nazwa, nie są już przypisane do istniejącego użytkownika. Jeżeli tak, serwis wstrzymuje proces rejestracji, zwracając odpowiedni kod błędu, który informuje, że proces rejestracji nie może zostać zakończony pomyślnie. Klasa posiada w sobie referencję do repozytorium użytkowników, dzięki czemu po walidacji danych wymaganych do utworzenia nowego użytkownika, serwis wywołuje metodę z repozytorium użytkowników, która zapisuje nowo utworzonego użytkownika.

4.17 Backlog Item 17

Tytuł zadania. Stworzenie funkcjonalności logowania użytkownika.

Opis zadania. Zaprogramowanie klasy do generowania tokenów uwierzytelniających, oraz dodanie do klasy UserService metody służącej do logowania.

Priorytet. Normalny

Definition of Done. W aplikacji stworzona jest klasa JwtHandler, wraz z metodą do generowania tokenów uwierzytelniających. Metoda do generowania tokenów przyjmuje jako parametry nazwę użytkownika oraz jego rolę. Metoda ta po zakończeniu swojego działania zwraca odpowiedni token, dzięki któremu użytkownik uwierzytelnia się do aplikacji. W klasie UserService stworzona zostaje metoda do logowania, która jako swoje parametry przyjmuje nazwę użytkownika oraz jego hasło. Metoda ta wykorzystuje repozytorium użytkowników do sprawdzenia czy dane podane przez użytkownika w procesie logowania są prawidłowe. Jeżeli dane są prawidłowe metoda wywołuje metodę generowania tokenów z klasy JwtHandler, dzięki czemu wygenerowany zostaje token uwierzytelniający. W przypadku niepowodzenia procesu logowania, tj. w przypadku podania błędnego hasła lub loginu, wygenerowania zostaje komunikat błędu informujący o podaniu błędnych danych do logowania.

4.18 Backlog Item 18

Tytuł zadania. Stworzenie funkcjonalności dodawania projektów.

Opis zadania. Stworzenie klasy Project, oraz ProjectService oraz dodawanie w niej metody służącej do dodawania nowych projektów.

Priorytet. Normalny

Definition of Done. W aplikacji stworzona jest klasa Project, która posiada następujące właściwości:

- Identyfikator projektu,
- Nazwa projektu,
- Data utworzenia projektu,
- Kontener z listą zadań projektu

Stworzona jest również klasa ProjectService, oraz metody która tworzy nowy obiekt Project na podstawie danych przesłanych przez użytkownika, a następnie korzystając z repozytorium projektów, wywołuje metodę dodającą projekt, która zapisuje nowo utworzony projekt w bazie danych.

4.19 Backlog Item 19

Tytuł zadania. Stworzenie funkcjonalności usuwania projektów.

Opis zadania. Stworzenie w klasie ProjectService metody służącej do usuwania projektu.

Priorytet. Normalny

Definition of Done. W klasie ProjectService dodana jest metoda, która usuwa podany projekt, na podstawie identyfikatora projektu przesłanego przez użytkownika. Metoda ta najpierw pobiera z repozytorium projektów obiekt projektu, wraz z zadaniami przypisanymi do tego projektu, a następnie przesyła ten projekt, do metody usuwającej projekt znajdującej się w klasie ProjectRepository. Metoda ta znajdująca się w ProjectRepository komunikuje się z bazą danych wywołując zapytanie usuwające projekt z bazy.

4.20 Backlog Item 20

Tytuł zadania. Stworzenie funkcjonalności modyfikowania projektu

Opis zadania. Dodanie do klasy ProjectService metody do modyfikacji projektu.

Priorytet. Niski

Definition of Done. W klasie ProjectService dodana jest metoda służąca do modyfikowania istniejącego wcześniej projektu. Metoda ta przyjmuje jako swoje argumenty identyfikator projektu, oraz nową nazwę projektu. Metoda za pomocą repozytorium projektów na podstawie przesłanego identyfikatora, pobiera projekt, zmienia, jego nazwę, a następnie wywołuje metodą zapisująca zmiany. W przypadku jeżeli do przesłanego przez użytkownika identyfikatora nie jest przypisany żaden projekt, metoda zwraca odpowiedni kod błędu informujący o niepowodzeniu aktualizacji projektu.

4.21 Backlog Item 21

Tytuł zadania. Stworzenie funkcjonalności dodawania nowych zadań.

Opis zadania. Stworzenie klasy Task będącej odwzorowaniem zadania, które może stworzyć użytkownik, oraz zaprogramowanie klasy TaskService i metody dodającej nowe zadanie.

Priorytet. Niski

Definition of Done. W aplikacji stworzona jest klasa Task, która posiada następujące pola:

- Identyfikator zadania
- Identyfikator projektu
- Identyfikator użytkownika
- Nazwa
- Opis
- Data dodania
- Termin realizacji
- Priorytet
- Etykieta

Zaprogramowana jest również klasa TaskService oraz metoda AddTask, która służy do tworzenia nowego zadania, na podstawie danych przesłanych przez użytkownika. Metoda ta przyjmuje jako parametry:

- identyfikator projektu
- identyfikator użytkownika
- nazwę zadania
- termin realizacji
- opis,
- priorytet

- etykietę

Metoda ta tworzy nowy obiekt klasy Task, a następnie używając Repozytorium Zadań, wywołuje metodę dodającą zadanie do bazy danych. Do odpowiedniego projektu(mającego identyfikator przesłany przez użytkownika) zostaje przypisane nowo utworzone zadanie.

4.22 Backlog Item 22

Tytuł zadania. Stworzenie funkcjonalności usuwania zadań.

Opis zadania. Stworzenie w klasie TaskService metody służącej do usuwania zadania.

Priorytet. Normalny

Definition of Done. W klasie TaskService dodana jest metoda, która usuwa podane zadanie, na podstawie identyfikatora zadania przesłanego przez użytkownika. Metoda ta najpierw pobiera z repozytorium zadań obiekt zadania, a następnie przesyła to zadanie, do metody usuwającej zadania znajdującej się w klasie TaskRepository. Metoda ta znajdująca się w TaskRepository komunikuje się z bazą danych wywołując zapytanie usuwające zadanie z bazy. Z projektu, do którego przypisane zostaje zadanie, zostaje ono usunięte.

4.23 Backlog Item 23

Tytuł zadania. Stworzenie funkcjonalności modyfikowania zadań.

Opis zadania. Dodanie do klasy TaskService metody do modyfikacji zadania.

Priorytet. Niski

Definition of Done. W klasie TaskService dodana jest metoda służąca do modyfikowania istniejącego wcześniej zadania. Metoda za pomocą repozytorium zadań na podstawie przesłanego identyfikatora, pobiera zadanie, a następnie wywołuje metodą zapisująca zmiany. W przypadku jeżeli do przesłanego przez użytkownika identyfikatora nie jest przypisany żadne zadanie, metoda zwraca odpowiedni kod błędu informujący o niepowodzeniu aktualizacji zadania. Zadanie, które użytkownik aktualizuje posiada przypisane nowe parametry.

4.24 Backlog Item 24

Tytuł zadania. Stworzenie funkcjonalności wysyłającej e–mail z zadaniami do wykonania bieżącego dnia.

Opis zadania. Zaprogramowanie w aplikacji klasy EmailNotifyService, która pobiera z repozytorium zadania do wykonania każdego dnia, a następnie wysyła je do użytkownika.

Priorytet. Niski

Definition of Done. Do użytkownika wysyłana jest codziennie na e–mail lista zadań do wykonania bieżącego dnia.

4.25 Backlog Item 25

Tytuł zadania. Stworzenie funkcjonalności pobierania listy zadań, do widoku kalendarza.

Opis zadania. Zaprogramowanie w klasach TaskService, oraz Repozytoriom Zadań, metod które będą pobierały listę zadań od bieżącego dnia do końca miesiąca. Lista tych zadań wykorzystywana będzie w widoku kalendarza.

Priorytet. Niski

Definition of Done. Do interfejsu użytkownika zwracana jest lista zadań, od dnia bieżącego do końca miesiąca.

4.26 Backlog Item 26

Tytuł zadania. Stworzenie funkcjonalności etykietowania zadań.

Opis zadania. Zaprogramowanie metody służącej do dodawania etykiet do zadań, dzięki czemu zadania będą mogły być grupowane np. w zależności od kontekstu wykonania(w domu, w pracy, itp.)

Priorytet. Niski

Definition of Done. Każde nowo utworzone zadanie może posiadać etykietę, określającą np. kontekst wykonania. W serwisie zadań zostaje dodana metoda, która modyfikuje istniejące zadanie nadając mu etykietę zadaną przez użytkownika.

4.27 Backlog Item 27

Tytuł zadania. Stworzenie repozytorium zadań

Opis zadania. Zaprogramowanie klasy repozytorium zadań.

Priorytet. Normalny

Definition of Done. Stworzona jest klasa Repozytorium Zadań. W repozytorium zaimplementowane są metody do dodawania, usuwania, pobierania zadania oraz aktualizacji jego danych. Repozytorium komunikuje się z silnikiem bazodanowym w celu trwałego zapisu danych na dysku.

5 Sprint 1

5.1 Cel

Stworzenie widoku i funkcjonalności rejestracji oraz logowania użytkownika.

5.2 Sprint Planning/Backlog

Tytuł zadania. Instalacja frameworka Angular.

- Estymata: XS

Tytuł zadania. Stworzenie interfejsu do rejestracji użytkownika.

- Estymata: XL

Tytuł zadania. Stworzenie interfejsu do logowania użytkownika.

- Estymata: XL

Tytuł zadania. Stworzenie repozytorium użytkowników.

- Estymata: S

Tytuł zadania. Stworzenie funkcjonalności rejestracji użytkowników.

- Estymata: L

Tytuł zadania. Stworzenie funkcjonalności logowania użytkowników.

- Estymata: XL

5.3 Realizacja

Tytuł zadania. Instalacja frameworka Angular.

Wykonawca. Anna Janusz

Realizacja. Instalacja przebiegła pomyślnie. Pliki zostały dodane do folderu `"...\src\MemoTime.UI"`. Czas instalacji nie odbiegł od określonej estymaty i zajął przyjęty czas 'XS'.

Tytuł zadania. Stworzenie interfejsu do rejestracji użytkownika.

Wykonawca. Anna Janusz

Realizacja. W frameworku Angular stworzone zostały moduły oraz komponenty do obsługi rejestracji użytkownika. Zostały podpięte pod główny moduł. Estymata była wprost-proporcjonalna do założeń.

Tytuł zadania. Stworzenie interfejsu do logowania użytkownika.

Wykonawca. Anna Janusz

Realizacja. W frameworku Angular stworzone zostały moduły oraz komponenty do obsługi rejestracji użytkownika. Zostały podpięte pod główny moduł. Estymata była mniejsza niż oczekiwana, gdyż komponent do logowania został opracowany na postawie modułu rejestracji.

Tytuł zadania. Stworzenie repozytorium użytkowników.

Wykonawca. Patryk Biel

Realizacja. Stworzone zastała klasa repozytorium, która zapisuje dane w pamięci. Implementacja repozytorium przebiegła bez problemów, a estymata była mniejsza niż oczekiwana

Tytuł zadania. Stworzenie funkcjonalności rejestracji użytkowników.

Wykonawca. Patryk Biel

Realizacja. W trakcie realizacji został stworzony endpoit do wysyłania żądań z warstwy frontowej. Endpoit wykorzystuje serwis, który parsuje dane otrzymane w żądaniu, a następnie zapisuje dane w repozytorium. Estymata czasowa zgodna z zakładaną.

Tytuł zadania. Stworzenie funkcjonalności logowania użytkowników.

Wykonawca. Patryk Biel

Realizacja. W aplikacji utworzony został endpoit do wysyłania danych do logowania. Po prawidłowym uwierzytelnieniu, do użytkownika zwracany jest token, który służy do autoryzacji do zasobów. Token ten należy umieszczać w nagłówku, poprzedzając go słowem kluczowym Bearer.
Przykład -¿ Authorization: Bearer Token
Mechanizm tworzenia tokenów oparty jest o Json Web Token, a algorytm służący do generowania tokento HMAC 256.

5.4 Sprint Review/Demo

«Sprawozdanie z przeglądu Sprint'u – czy założony cel (przyrost) został osiągnięty oraz czy wszystkie zaplanowane Backlog Item'y zostały zrealizowane? Demonstracja przyrostu produktu».

6 Sprint 2

6.1 Cel

«Określić, w jakim celu tworzony jest przyrost produktu».

6.2 Sprint Planning/Backlog

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach")».

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach")».

«Tutaj dodawać kolejne zadania»

6.3 Realizacja

Tytuł zadania. «Tytuł».

Wykonawca. «Wykonawca».

Realizacja. «Sprawozdanie z realizacji zadania (w tym ocena zgodności z estymatą). Kod programu (środowisko `verbatim`):

```
for (i=1; i<10; i++)
...
```

».

Tytuł zadania. «Tytuł».

Wykonawca. «Wykonawca».

Realizacja. «Sprawozdanie z realizacji zadania (w tym ocena zgodności z estymatą). Kod programu (środowisko `verbatim`):

```
for (i=1; i<10; i++)
...
```

».

«Tutaj dodawać kolejne zadania»

6.4 Sprint Review/Demo

«Sprawozdanie z przeglądu Sprint'u – czy założony cel (przyrost) został osiągnięty oraz czy wszystkie zaplanowane Backlog Item'y zostały zrealizowane? Demostracja przyrostu produktu».

«Tutaj dodawać kolejne Sprint'y»

Literatura

[1] S. R. Covey, *7 nawyków skutecznego działania*, Rebis, Poznań, 2007.

[2] Tobias Oetiker i wsp., Nie za krótkie wprowadzenie do systemu LaTeX 2_ε, `ftp://ftp.gust.org.pl/TeX/info/lshort/polish/lshort2e.pdf`

[3] K. Schwaber, J. Sutherland, *Scrum Guide*, `http://www.scrumguides.org/`, 2016.

[4] `https://agilepainrelief.com/notesfromatooluser/tag/scrum-by-example`

[5] `https://www.tutorialspoint.com/scrum/scrum_user_stories.htm`