

Event Storming

Wstęp

Występuje maksymalnie na 3 poziomach

1. Event Storming **Big Picture**
2. Event Storming **Process Level**
3. Event Storming **Design Level** (czasem wchłania Process Level i wtedy mamy 2 poziomy)

Tworzy modele procesów biznesowych. Taki model to grupa uporządkowanych zdarzeń domenowych

Pomaga określić granice (zidentyfikować) **Subdomen, Bounded Contextów, Agregatów**

Dostarcza modeli do implementacji **modelu domenowego** w kodzie

Stanowi podstawę do **decyzji architektonicznych** i podziału systemu na **komponenty**

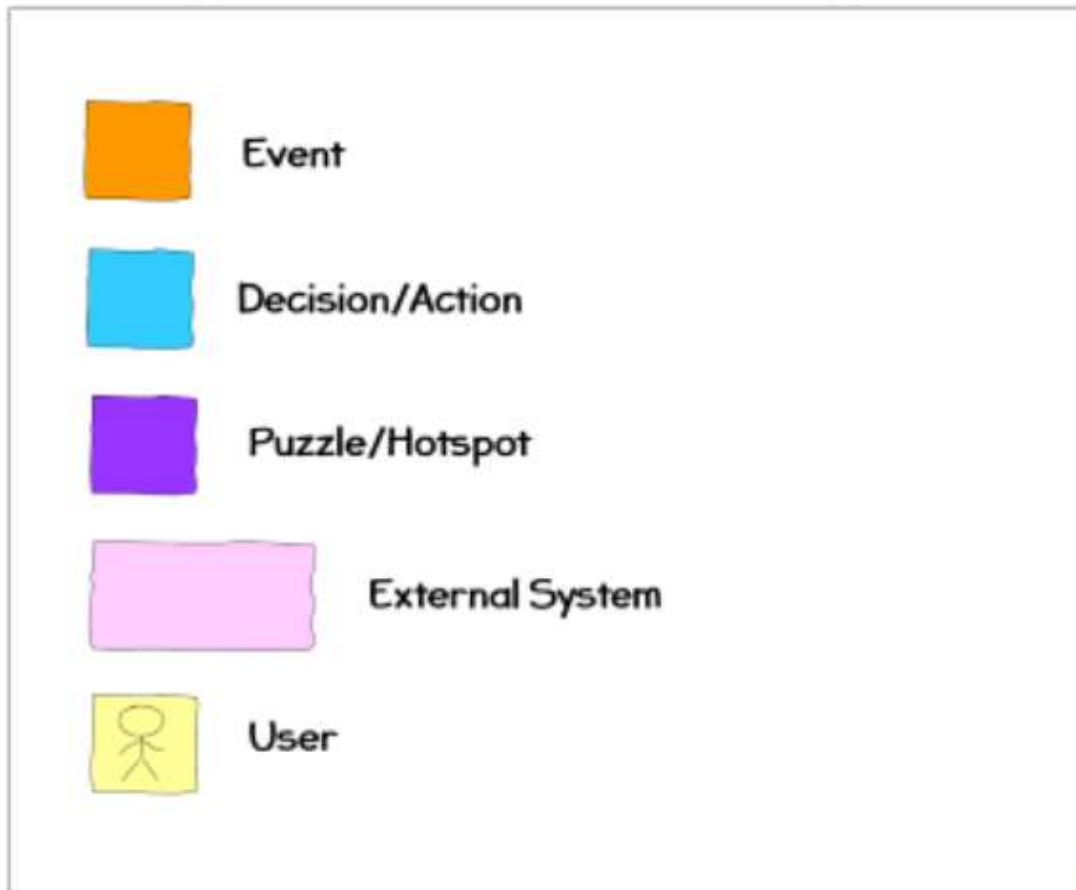
Event Storming Big Picture

Piszemy i komunikujemy się według **Ubiquitous Language** (wszechobecny język subdomen)

1. **Big Picture**
 - Zaczynamy od kluczowych zdarzeń, wokół nich dodajemy pozostałe zdarzenia
 - Piszemy **Zdarzenia**, jeśli coś jest niejasne piszemy **Hotspoty**
 - Wyznaczymy **Subdomeny** lub od razu **Bounded Contexty** według pogrupowania modeli procesów biznesowych
2. Dodajemy **Komendy, Aktorów, Systemy Zewnętrzne**

Jak jest składnia?

Big Picture Event Storming



Czym jest Domain Event?



Domain
Event

- Zdarzenie Domenowe opisuje istotną zmianę stanu procesu lub systemu
- Zdarzenie jest jednoznacznie identyfikowalne, można określić zarówno jego nazwę, jak i moment w czasie czy właściwości
- Zdarzenie jest źródłem cennych informacji i stanowi świetny punkt wyjściowy w rozmowie z ekspertem domenowym

- Istotne zdarzenie biznesowe w procesie biznesowym (np. Wystawiono fakturę)
- Czas przeszły (tryb temporalny, coś się stało więc ktoś to zrobił, coś ma jakiś stan i ma poprzednika / następnika i swoje konsekwencje w czasie)

Czym jest Command?



- Rozkaz opisuje żądanie zmiany w procesie lub systemie, wyzwolone przez np. aktora, upływający czas lub inny system
- Rozkaz jest jednoznacznie identyfikowalny, posiada nazwę oraz szereg właściwości
- W wyniku przetworzenia Rozkazu może pojawić się jedno lub więcej Zdarzeń Domenowych

- Np. Wystaw fakturę

Kogo zaprosić na warsztat?

- Klient: Eksperti domenowi, Osoby decyzyjne
- My: Facylitator, Developerzy, Zespół projektowy

Jak Przygotować się do warsztatu Event Storming Big Picture?

Przyda się

- Przestrzeń
- Ściany - papier plakatowy, taśma malarska
- Karteczki i Flamastry
- Flipchart
- Minutnik

- Narzędzie do dokumentacji miro board – można na nim przeprowadzić Event Storming Zdalny, czasem jedyne wyjście, choć wskazany jest Event Storming Big Picture wspólnie, nie zdalnie.

Jakie są fazy Warsztatu?

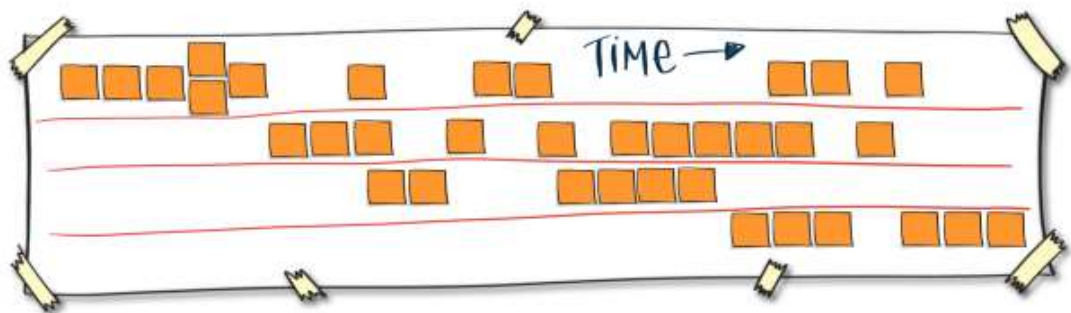
Faza 0 - Rozpoczęcie (przedstaw cel spotkania)

Faza 1 - Chaotyczna eksploracja

- Każdy osobno klei zdarzenia - opcjonalne, ale się sprawdza
- Kończymy chaotyczną eksplorację - kiedy pierwsza osoba skończy wypływać eventy (karteczki)

Faza 2 - Wprowadzenie osi czasu - uporządkowanie eventów zgodnie z linią czasu

- Wprowadzenie struktury
- Usunięcie duplikatów
- **Hotspot** - czerwona karteczka - jest jakiś problem ogarniemy go później
- SwimLine - bardzo dużo zdarzeń i może pojawić się sporo równoległych procesów



Faza 3 - Opowieść od końca

- Angażuje Facylitatora
- Debugowanie procesu biznesowego
- Znajdowanie brakujących zdarzeń - biznes ma się tu skupić (wejść w stan skupienia pomaga im przejście od końca)
- Co się dzieje
 - Nowe zdarzenia
 - Poprawki
 - Rozgałęzienia
- Pytania Facylitatora by pomoc debugować proces biznesowy
 - Wcześniej

- Co jeszcze musiało się stać, żeby X
- Pomiedzy
 - Czy między X a Y dzieje się coś jeszcze?
- Alternatywa
 - Co, jeśli X by się nie powiodło
- Strukturyzowanie układu zdarzeń przez
 - Klejenie w dół (eventy mogą zadziać się wszystkie, bądź minimum jeden w dowolnej kolejności)
 - Alternatywa krótka
 - Alternatywa długa
 - Po skosie w dół (automatyczne następstwa event-ów)

Faza 4 – Dodajemy **komendy**, zastanawiamy się i oznaczamy kto jest inicjatorem komendy: **Aktor**, **Zewnętrzny system**, **Czas**, Jakaś procedura?

- Dodawanie komend i aktorów prowokuje do dyskusji, dlaczego użytkownik wykonał jakieś kroki. Niektóre kroki są trywialne, inne jednak prowokują do dyskusji w stylu „Dlaczego użytkownik X aktywował usługę”, co prowadzi do podważania wcześniejszych założeń.
- Komendy pomagają nam zidentyfikować Read Modele – bo zastanawiamy się np. na podstawie jakich danych użytkownik podjął decyzję
- Zdarzenia mogą być rozpoczynane przez komendy, lecz również przez czas, zewnętrzne eventy, kaskadę eventów, dokumenty
- Nie każde zdarzenie musi mieć aktora

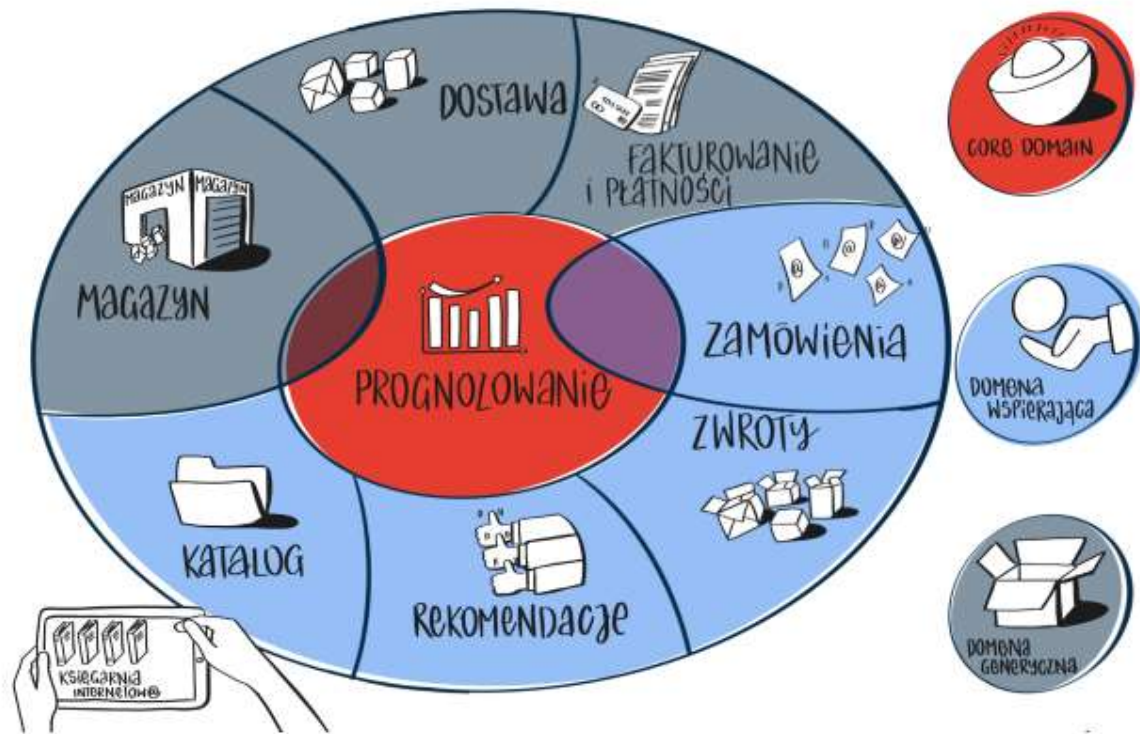
Faza 5 - Problemy i okazje

- **Problem** - fioletowa karteczka
- **Okazja** - zielone karteczka

Faza 6

- **Czarnymi strzałkami** pokazujemy najważniejsze **problemy** / **okazje** / **informacje** i tam chcemy usprawnić lub utworzyć proces (na design level es), możliwe, że tam jest **Core Domain**

Odkrywanie Core Domain i Subdomain



Rysunek 1 Źródło: DNA

Rodzaje Subdomain

- **Core Domain** (Najważniejsza, Przewaga rynkowa, najlepszy kod - Taktyczne DDD)
- **Supportive Subdomain** (Nie istnieje gotowy produkt je dostarczający)
- **Generic Subdomain** (Istnieje gotowy produkt je dostarczający)

Odkrywanie Subdomen można zrealizować według poniższych heurystyk (wskazówek)

- Struktura organizacji - podział wg struktury dobra heurystyka na początek
- Różni eksperci domenowi
- Język ekspertów domenowych - np. Książka znaczy co innego w różnych obszarach, więc różne subdomeny

- Patrz na wartość biznesową - coś co daje dużą wartość biznesową jest kandydatem na Subdomain a nawet Core Subdomain
- Patrz na kroki procesu biznesowego
- Pamiętaj, że subdomeny się zmieniają - patrz czy nasz model mentalny podąża, za modelem operacyjnym

Event Storming Process Level

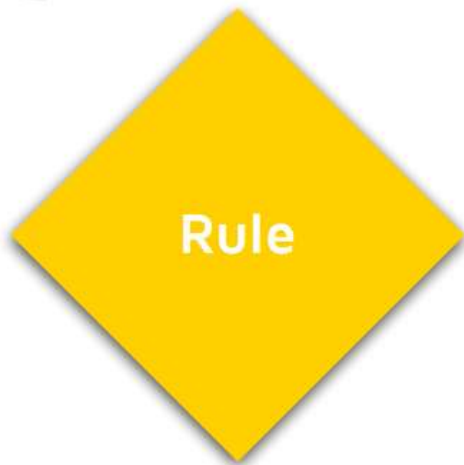
W ramach ES Process Level jesteśmy już w **przestrzeni rozwiązania** (wcześniej była to **przestrzeń problemu / domeny**).

Po Event Storming na poziomie **Big Picture** możemy kontynuować Event Storming na poziomie **Process Level**. W celu:

- Identyfikacji **Reguł Biznesowych** (potrzebni Eksperci Domenowi)
- Rozwiązania **Hot Spot-ów**
- Nazwania **Procesów Biznesowych**
- Spojrzeniu na zidentyfikowane w Big Picture **Szanse i Ryzyka** – określamy dla nich **Drivery i Metryki i proponujemy rozwiązania na modeleach procesów biznesowych**
- Dodaniu **Polityk** (Polityka- obsługa długo trwającego asynchronicznego procesu)
- Identyfikacji **Bounded Context-ów** i ich granic. Lub ponownej weryfikacji i polepszeniu granic określeniu w Big Picture



Czym jest Rule?



- Reguła opisuje wymagany warunek spójności
- Posiadaczami informacji o regułach są zazwyczaj eksperci domenowi
- Zapewnienie spójności reguł biznesowych jest zadaniem realizowanym przez Agregaty

- Reguła znajduje się między komendą a zdarzeniem.
- Np. Wystaw fakturę (Komenda) → Jeśli kwota mniejsza od < 10000 (Reguła) → Wystawiono fakturę (Zdarzenie)

Utworzenie Bounded Context-ów i ich granic

W tym momencie przenosimy się z **przestrzeni problemu** (domeny biznesu) do **przestrzeni rozwiązania**

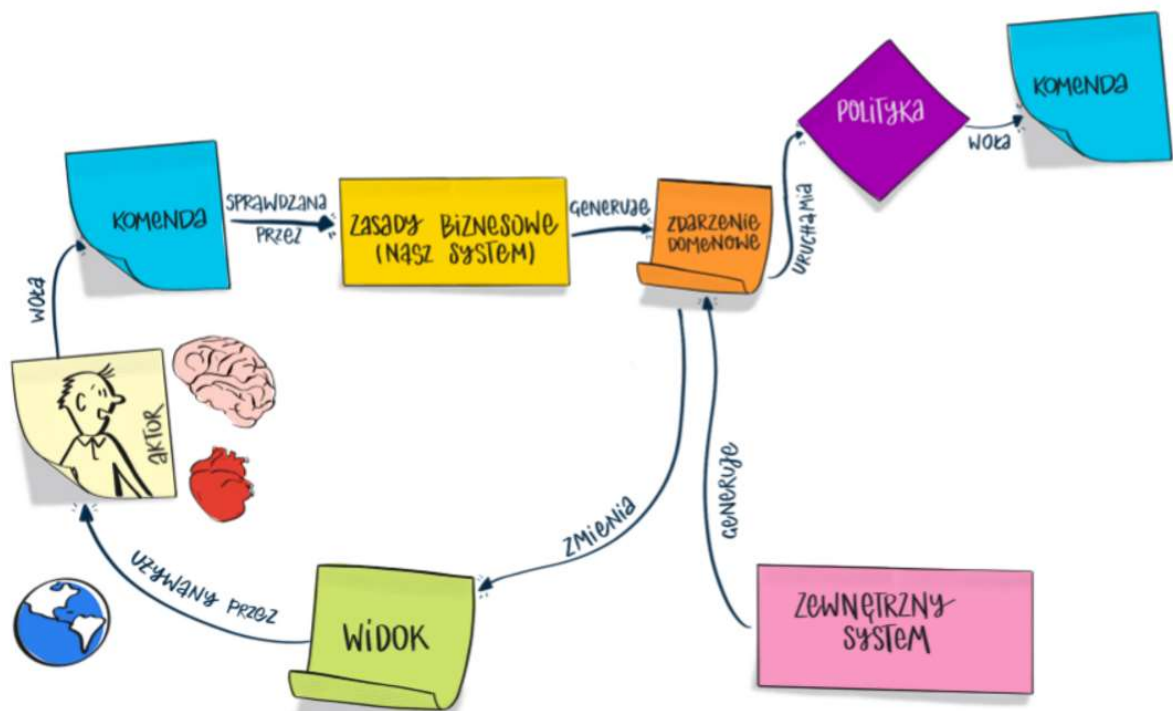
Bounded Context znajduje się już w świecie oprogramowania. Subdomeny są w świecie domeny biznesu. Bounded Contexty są często po prostu odzwierciedleniem subdomen, ale w funkcjonują w przestrzeni oprogramowania. Po ich wydzieleniu mapują się one nam na komponenty w architekturze naszego systemu. Używamy w nich tzw. **Ubiquitous Language (Wszechobecny / Jednolity Język** w ramach subdomeny, do której należą). Granice Bounded Context-ów identyfikujemy za pomocą heurystyk. A „na chłopski rozum” określają je przede wszystkim sensownie pogrupowane modele procesów biznesowych, które znajdują się na boardzie event stormingowym.

Metafora różnicy między Subdomenami i bounded Contextami to: Do pokoju do podłogi (Subdomena) dodajemy dywan (Bounded Context)

Heurystyki walidowania granic kontekstów i tym samym odkrywania bounded contextów

1. **Autonomia kontekstu** – Bounded Context ma być jak najbardziej autonomiczny
 - Czy nasz Bounded Context może sam podejmować decyzje?
 - Czy musi pytać o zgodę kilka innych Bounded Context?
 - Odpowiada na główne pytanie biznesowe?
 - Zasada pojedynczej odpowiedzialności
2. Liczba kontekstów w procesie biznesowym - im **mniej** Bounded Context uczestniczy w danym procesie biznesowym (proces biznesowy jest oznaczony na boardzie Event Storming-owym) tym pewnie **lepiej**
 - Jak wiele kontekstów uczestniczy w danym procesie biznesowym?
3. Szukaj informacji zmieniających się razem
4. Szukaj informacji używanych razem
5. Zadaj sobie pytania
 - Jaka jest odpowiedzialność kontekstu? - jeśli kontekst robi zbyt dużo to może trzeba go rozbić na mniejsze
 - Ile integracji ma kontekst?
 - Czy jest jedno źródło prawdy każdej informacji? - czy każda informacji ma właściciela - kontekst
 - Czy istnieje schizofreniczny kontekst?
 - Np. sprawdza czy klient indywidulany / korporacyjny -> najprawdopodobniej wydzielić 2 konteksty
6. Szukaj anty-wymagań
 - Anty-wymagania to biznesowe reguły nie mające sensu
 - Pomagają przy walidacji granic

Event Storming Design Level



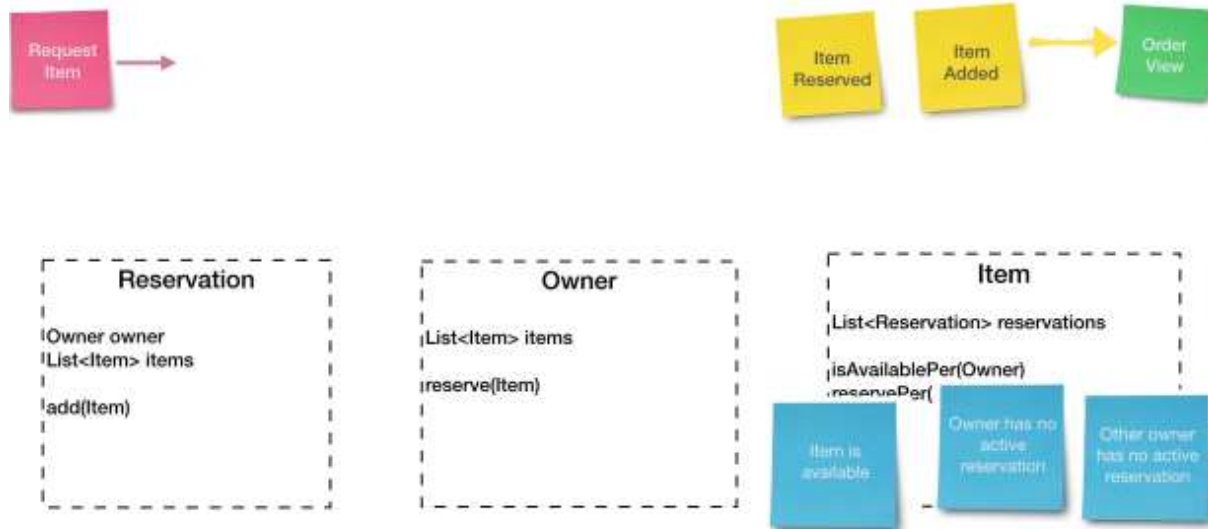
Źródło: leanpub/introducing_event_storming

Rysunek 2 Współgrające ze sobą elementy Event Stormingu

W tej fazie poprawiamy nasze modele i zastanawiamy się nad decyzjami architektonicznymi. Przede wszystkim w tej fazie określam **Agregat** - klasa, która trzyma te dane i te **Reguły**, które muszą być spójne atomowo

Agregat - model reguł, które muszą być spójne synchronicznie

Availability Context



Rysunek 3 Proces tworzenia Agregata

Podsumowanie

Event Storming wspiera

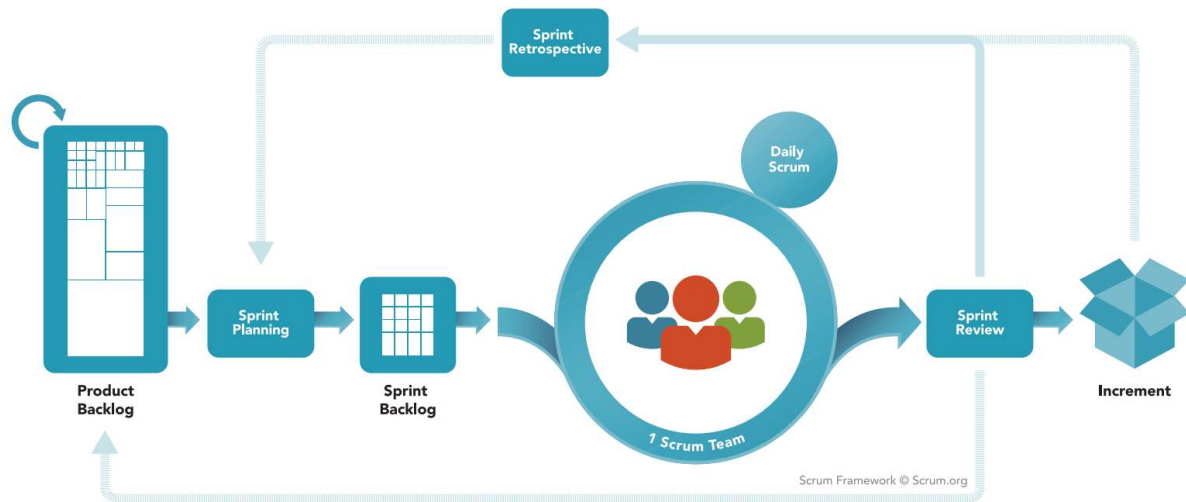
- Eksploracje przestrzeni problemu (domeny biznesu, do której tworzymy oprogramowanie) poprzez dokumentację procesów biznesowych w zidentyfikowanych subdomenach
- Identyfikację reguł biznesowych, ulepszenie procesów biznesowych, identyfikację bounded contextów tworzonego systemu
- Utworzenie odpowiednich komponentów architektury (na podstawie granic bounded contextów)
- Utworzenie Agregatów (model w Event Storming Design Level)
- Implementację Sagi (jej modelem jest polityka)
- Implementację CQRS-a (jego modelem jest Command i Read Model)
- Implementację odpowiedniej komunikacji między komponentami (z modeli widać, czy mamy do czynienia z komunikatem typu query, command czy event)

Uproszczoną alternatywą do Event Storming jest www.eventmodeling.org

SCRUM

Wsadem do prowadzenia projektu w metodyce **SCRUM** są: Boardy z sesji Event Storming Big Picture, Process Level, Design Level. A w szczególności odkryte Subdomain-y, stworzone Bounded Contexty, nazwane procesy biznesowe, modele procesów biznesowych, sposoby komunikacji wewnątrz bounded contextów i między nimi

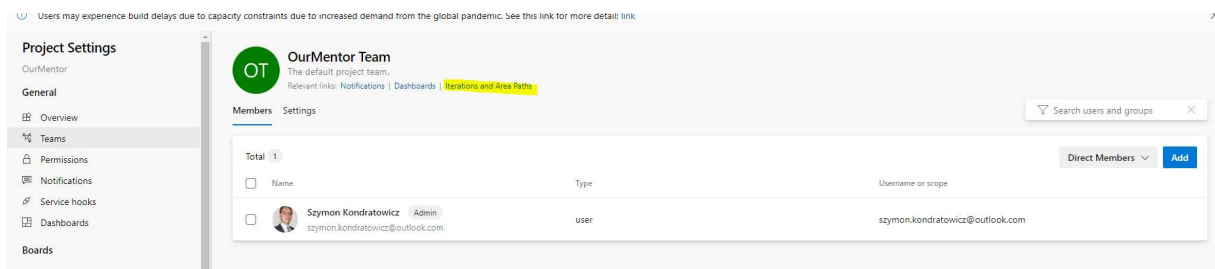
SCRUM FRAMEWORK



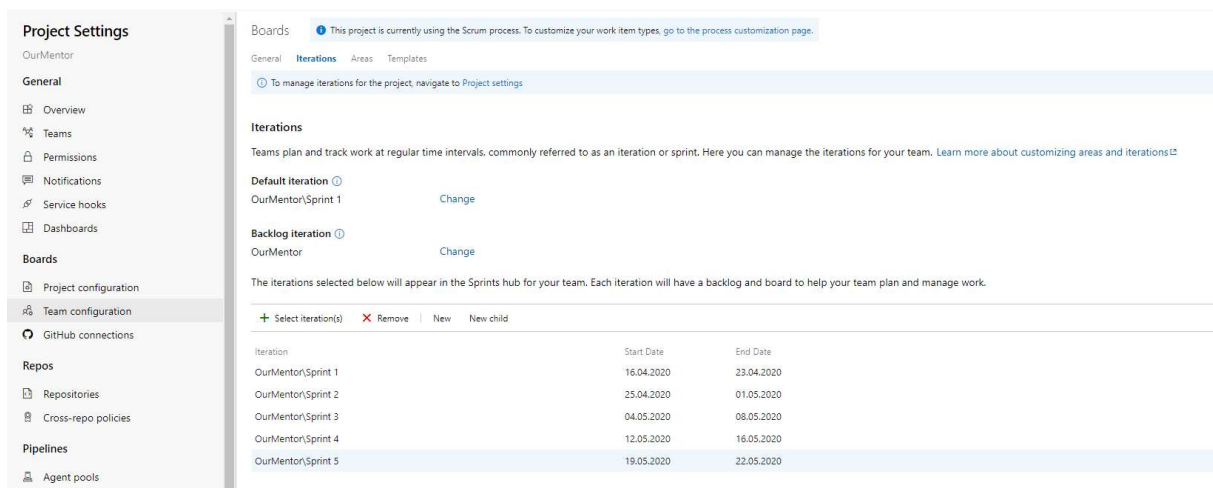
Azure DevOps Boards

- Planowanie, Zarządzanie i Monitorowanie pracy całego Zespołu
- Product Backlog, Sprint Backlog, Task Boards

Praca z Zespołami (Teams), Obszarami (Areas) i Iteracjami (Iterations)



Rysunek 4 zdefiniowanie zespołu i areas-ów

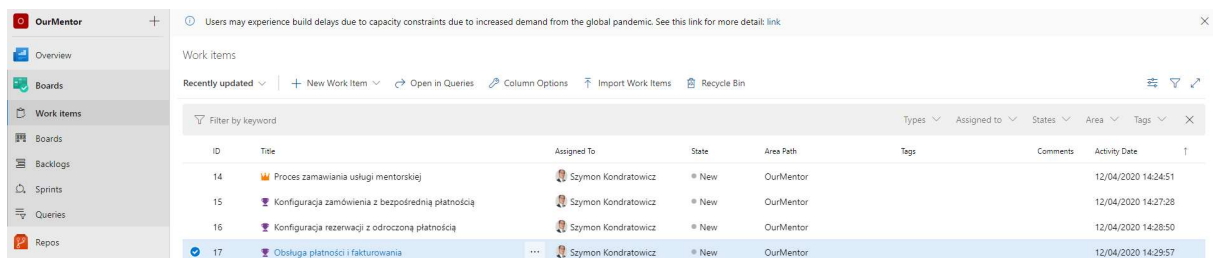


Rysunek 5 Zdefiniowanie sprintów

Praca z Work Item-ami na podstawie wyniku event Storming

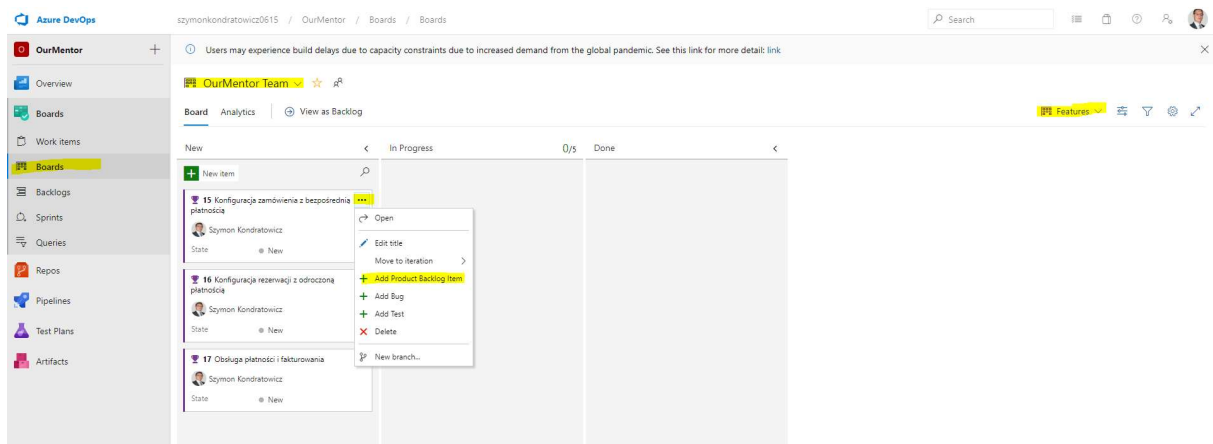
Na podstawie Event Storming mamy wsad do tworzenia poszczególnych Work Item-ów, można zaproponować poniższe mapowanie (work item – element z event sotrmimg)

- **Epic** – Bounded Context
- **Feature** – nazwa Procesu Biznesowego w Bounded Context
- **Product Backlog Item / User Story**
 - Wynika z komendy i zdarzenia (można uwzględnić Aktora, Politykę, Integracja, Readmodel)
 - Opis sprzed wykonania zdarzenia, jeśli można to z aktorem. Np. Jako klient mogę wybrać jedną lub wiele usług.
 - Jeśli zdarzenie jest automatycznym następstwem, to scalamy taki ciąg zdarzeń w jeden Product Backlog Item
- Product Backlog Item – dzielimy na **Taski** techniczne
 - **Frontend, Endpoint, Logika domenowa, Persystencja, Integracja synchroniczna** lub **asynchroniczna** przez **Politykę** (którą implementuje wzorzec **Saga** lub **Process Manager**)
 - Taski techniczne najlepiej robić po ES Design Level – widać wtedy elementy modelu technicznego



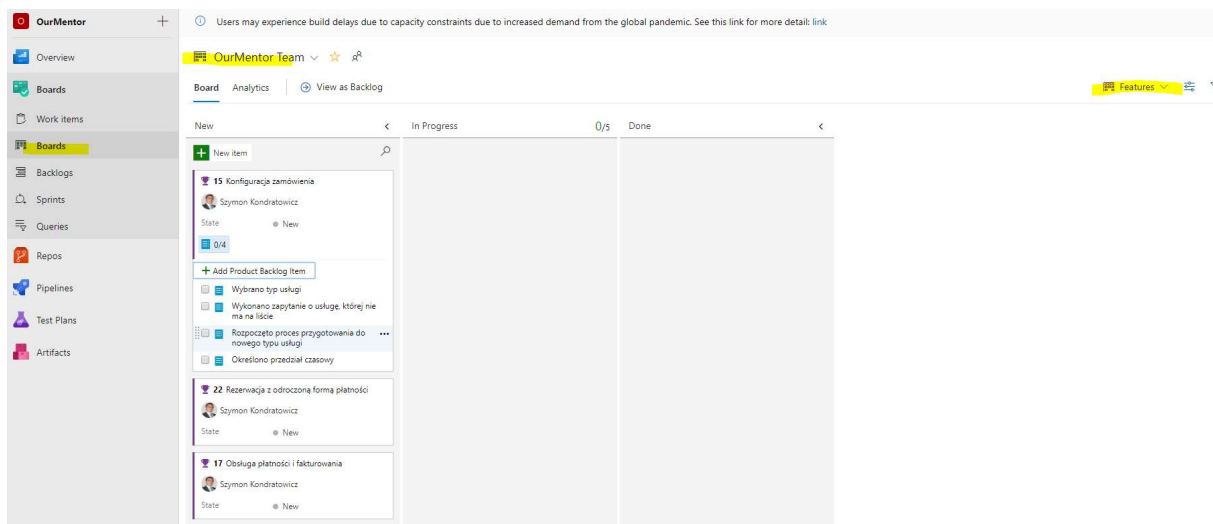
ID	Title	Assigned To	State	Area Path	Tags	Comments	Activity Date
14	Proces zamawiania usługi mentorskiej	Szymon Kondratowicz	New	OurMentor			12/04/2020 14:24:51
15	Konfiguracja zamówienia z bezpośrednią płatnością	Szymon Kondratowicz	New	OurMentor			12/04/2020 14:27:28
16	Konfiguracja rezerwacji z odroczoną płatnością	Szymon Kondratowicz	New	OurMentor			12/04/2020 14:28:50
17	Obsługa płatności i fakturowania	Szymon Kondratowicz	New	OurMentor			12/04/2020 14:29:57

Rysunek 6 Utworzenie Epic-a i zlinkowanych relacją child - parent Feature-ów



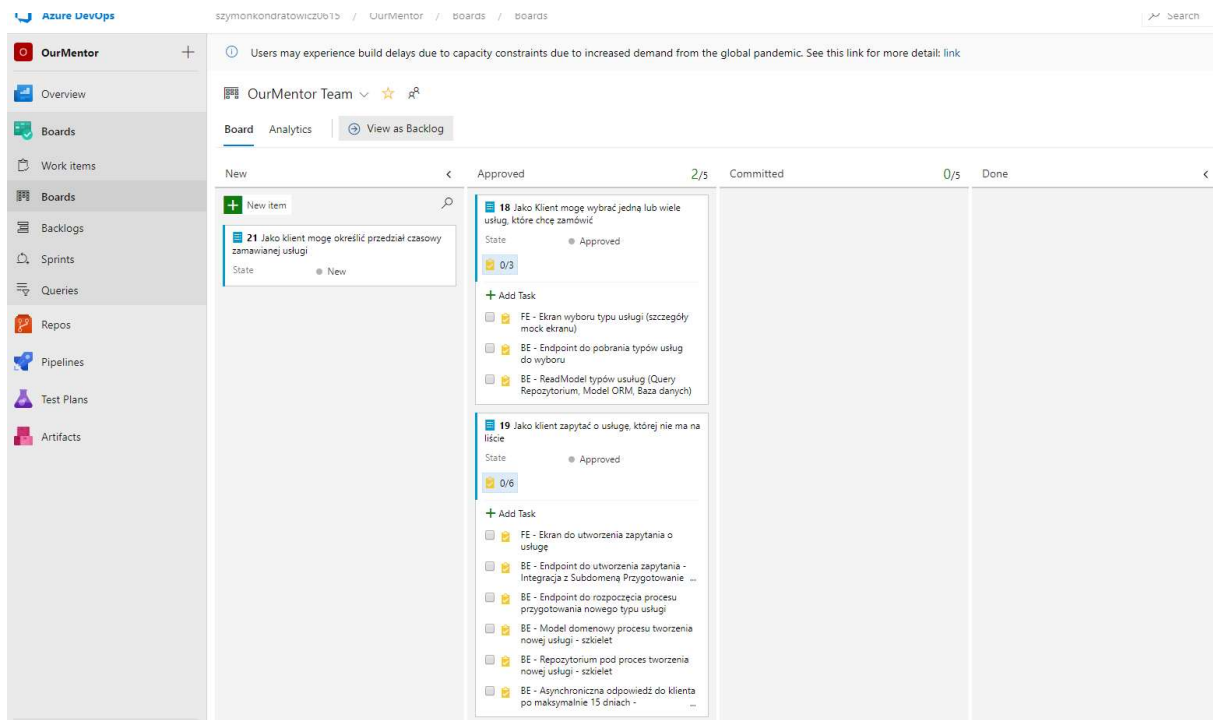
The screenshot shows the Azure DevOps Boards interface. The left sidebar contains navigation links for Overview, Boards, Work Items, Backlogs, Sprints, Queries, and Repos. The main area displays a Kanban board for the 'OurMentor Team'. The board has columns for 'New', 'In Progress', and 'Done'. Item 15, 'Konfiguracja zamówienia z bezpośrednią płatnością', is in the 'New' column. A context menu is open over this item, showing options: 'Open', 'Edit title', 'Move to iteration', 'Add Product Backlog Item' (highlighted), 'Add Bug', 'Add Test', 'Delete', and 'New branch...'. Item 16, 'Konfiguracja rezerwacji z odroczoną płatnością', and item 17, 'Obsługa płatności i fakturowania', are also in the 'New' column.

Rysunek 7 Utworzenie Product Backlog Item-ów



The screenshot shows the Azure DevOps Boards interface. The left sidebar contains navigation links for Overview, Boards, Work Items, Backlogs, Sprints, Queries, and Repos. The main area displays a Kanban board for the 'OurMentor Team'. The board has columns for 'New', 'In Progress', and 'Done'. The 'New' column contains three items: 15, 22, and 17. Item 15, 'Konfiguracja zamówienia z bezpośrednią płatnością', is expanded, showing a checklist with tasks: 'Wybrano typ usługi', 'Wykonano zapytanie o usługę, której nie ma na liście', 'Rozpoczęto proces przygotowania do nowego typu usługi', and 'Określono przedział czasowy'. Item 22, 'Rezerwacja z odroczoną formą płatności', and item 17, 'Obsługa płatności i fakturowania', are also in the 'New' column.

Rysunek 8 Utworzenie Product Backlog Item-ów



Rysunek 9 Dodanie tasków do PBI-sów

Zarządzanie Sprintami i Capacity

Planowanie

Tworzenie **Sprint Backlog**-a podczas **Planowania Sprintu** – typowo pierwszy dzień **Sprintu**. Każdy Sprint ma określony czas trwania (z reguły 2 tygodnie). Podczas planowania **Product Owner** identyfikuje **Product Backlog Item**-y, które powinny być zrealizowane w trakcie sprintu. Product Backlog Itemy mają pole **Priority**, które stanowi o ich **ważności** i **kolejności** wykonania.

1 faza planowania sprintu

Zidentyfikowanie, które Product Backlog Itemy wejdą do Backlog sprintu, a które przesuwamy na następne sprinty. Product Backlog Item-y mają pole Effort, które można wypełnić np. elementem ciągu Fibonacciego. Pozwala wycenić czasochłonność. Pole te uzupełniamy na przykład w **SCRUM**

Pokerze.

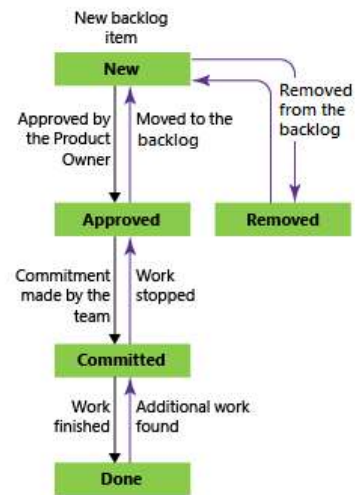
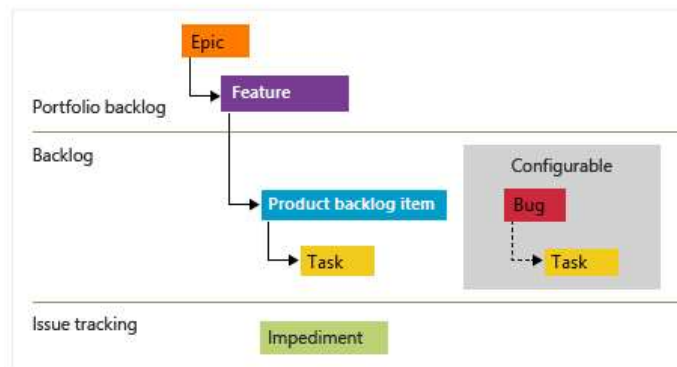
2 faza planowania

Identyfikacja i wycena tasków, wycena już może nastąpić w godzinach, robi ją zespół developerski, można również użyć SCRUM Poker-a.

3 faza planowania sprintu

Sprawdzenie **Capacity** zespołu po wycenie i ewentualne przesunięcie PBI-sów z Task-ami na przyszły sprint, lub dobranie nowych.

Po planowaniu następuję **praca nad Task-ami**. Według **flow przejść**. Rysunek niżej. Przydatną opcją jest dodanie kolumn: **Code Review** i **QA Tested**. Po zweryfikowaniu kodu w code review i przetestowaniu zostaje oddany na release branche-a.



Rysunek 10 SCRUM Proces w AzureDevops

Rysunek 11 Backlog – wykorzystany do planowania sprint - Przesunięcie na przyszły sprint

OurMentor

+

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Repos

Pipelines

Test Plans

Artifacts

Users may experience build delays due to capacity constraints due to increased demand from the global pandemic. See this link for more detail: link

OurMentor Team

Taskboard

Backlog

Capacity

Analytics

+ New Work Item

Column Options

...

Order	Title	State	Assigned To	Rema...
1	Jako Klient mogę wybrać jedną lub wiele usług, które ch...	Committed		24
	FE - Ekran wyboru typu usługi (szczegóły mock ekranu)	To Do	Szymon Kond...	8
	BE - Endpoint do pobrania typów usług do wyboru	To Do		8
	BE - ReadModel typów usług (Query Repozytorium, Mo...	To Do		8
2	Jako klient zapytać o usługę, której nie ma na liście	Committed		68
	FE - Ekran do utworzenia zapytania o usługę	To Do		8
	BE - Endpoint do utworzenia zapytania - Integracja z Sub...	To Do		4
	BE - Endpoint do rozpoczęcia procesu przygotowania no...	To Do		8
	BE - Model domenowy procesu tworzenia nowej usługi - ...	To Do		16
	BE - Repozytorium pod proces tworzenia nowej usługi - s...	To Do		16
	BE - Asynchroniczna odpowiedź do klienta po maksymaln...	To Do		16

Rysunek 12 Sprint Backlog

OurMentor

Overview

Boards

Work items

Boards

Backlogs

Sprints

Queries

Repos

Pipelines

Test Plans

Artifacts

Project settings

szymonkondratowicz0615 / OurMentor / Boards / Sprints

Search

OurMentor Team

16 kwietnia - 23 kwietnia

6 work days

Taskboard

Backlog

Capacity

Analytics

+ New Work Item

Column Options

Collapse all

To Do 92 h

In Progress

Done

18 Jako Klient mogę wybrać jedną lub wiele usług, które chcę zamówić

Szymon Kondra...

24 h

State

Committed

23 FE - Ekran wyboru typu usługi (szczegóły mock ekranu)

Szymon Kondra...

8

State

To Do

24 BE - Endpoint do pobrania typów usług do wyboru

Szymon Kondra...

8

State

To Do

28 BE - ReadModel typów usług (Query Repozytorium, Model CRM, Baza danych)

Szymon Kondra...

8

State

To Do

19 Jako klient zapytał o usługę, której nie ma na liście

Szymon Kondra...

68 h

State

Committed

25 FE - Ekran do utworzenia zapytania o usługę

Szymon Kondra...

8

State

To Do

26 BE - Endpoint do utworzenia zapytania - Integracja z Subdomeną

Szymon Kondra...

4

State

To Do

27 BE - Endpoint do rozpoczęcia procesu przygotowania nowego typu

Szymon Kondra...

8

State

To Do

29 BE - Model domenowy procesu tworzenia nowej usługi - szkieleł

Szymon Kondra...

16

State

To Do

30 BE - Repozytorium pod proces tworzenia nowej usługi - szkieleł

Szymon Kondra...

16

State

To Do

31 BE - Asynchroniczna odpowiedź do klienta po maksymalnie 15 dniach -

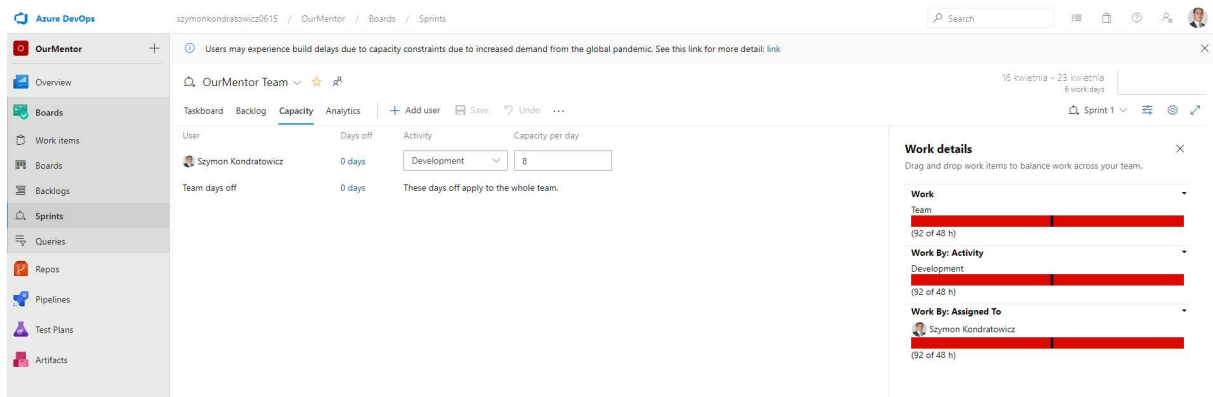
Szymon Kondra...

16

State

To Do

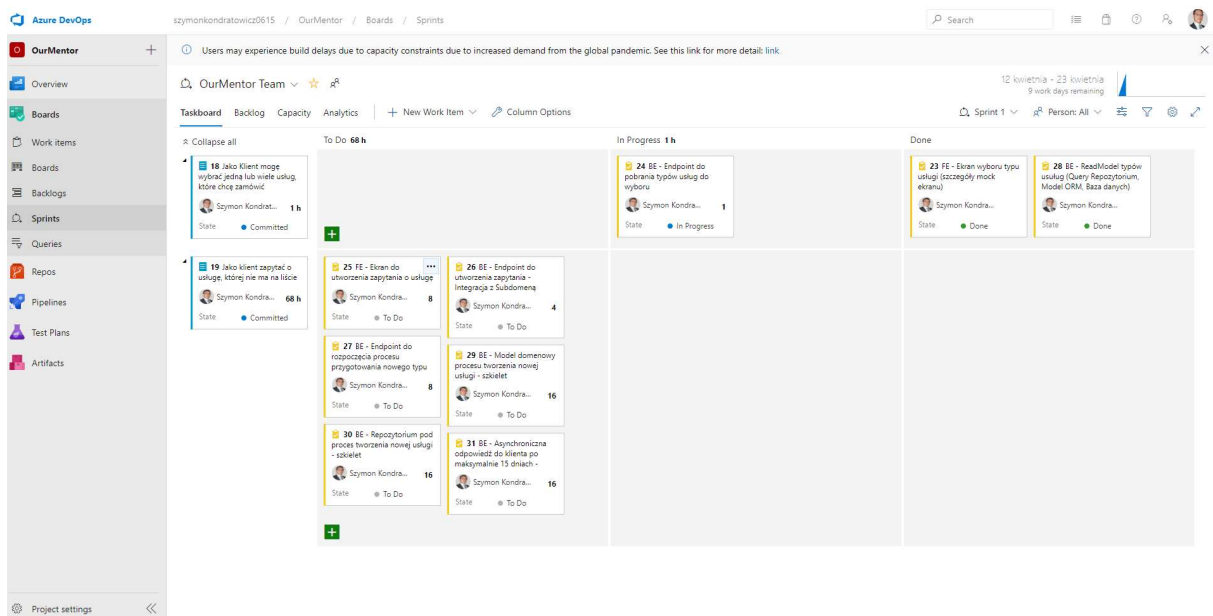
Rysunek 13 Sprint Task Board



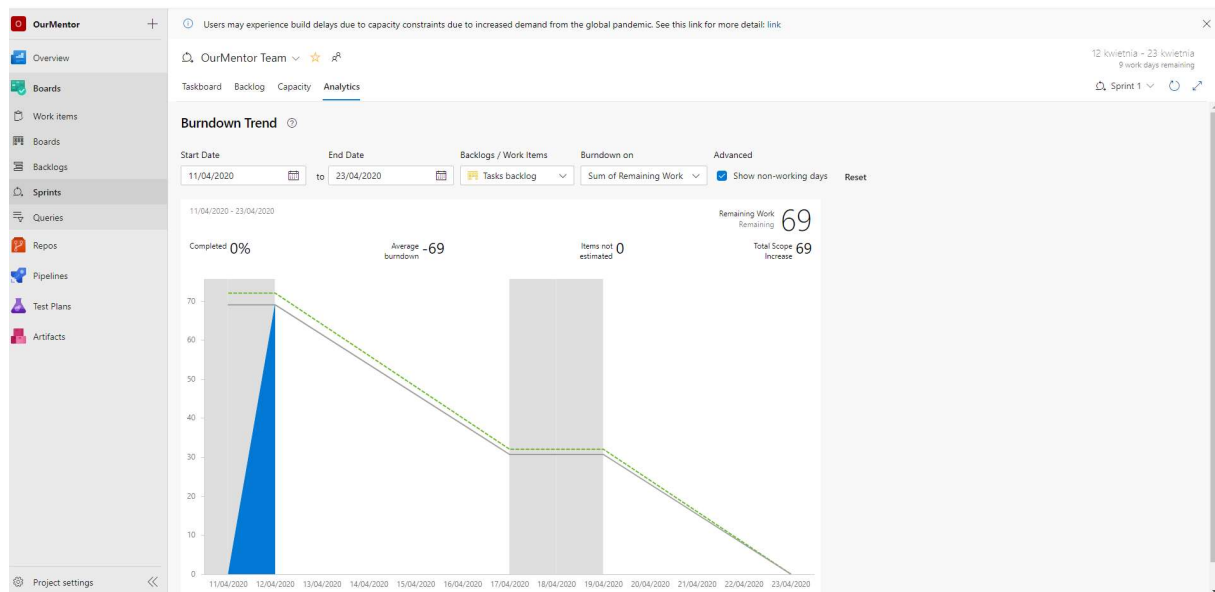
Rysunek 14 Sprint Capacity - analiza obciążenia pracą zespołu

Daily Scrum

Co zrobiłem? Co robię? Co będę robił? Jakie mam problemy? Na każde pytanie odpowiada każdy na widoku Sprint TaskBoard-a, ewentualnie analiza Burndown chart-a



Rysunek 15 Sprint Task Board, na którym realizujemy Daily



Rysunek 16 Burndown chart do oceny postępów pracy

Sprint review and retrospective

- <https://www.scrum.org/resources/what-is-a-sprint-review>
- <https://www.scrum.org/resources/what-is-a-sprint-retrospective>

Źródła

- szkolenie DNA
- szkolenie mikroserwis.net
- Introducing EventStorming, Alberto Brandolini
- Applying Domain Driven Design with CQRS and Event_Sourcing, Nick Chamberlan
- <https://docs.microsoft.com/pl-pl/azure/devops/?view=azure-devops>
- <https://www.azuredevopslabs.com/labs/azuredevops/agile/>
- scrum.org