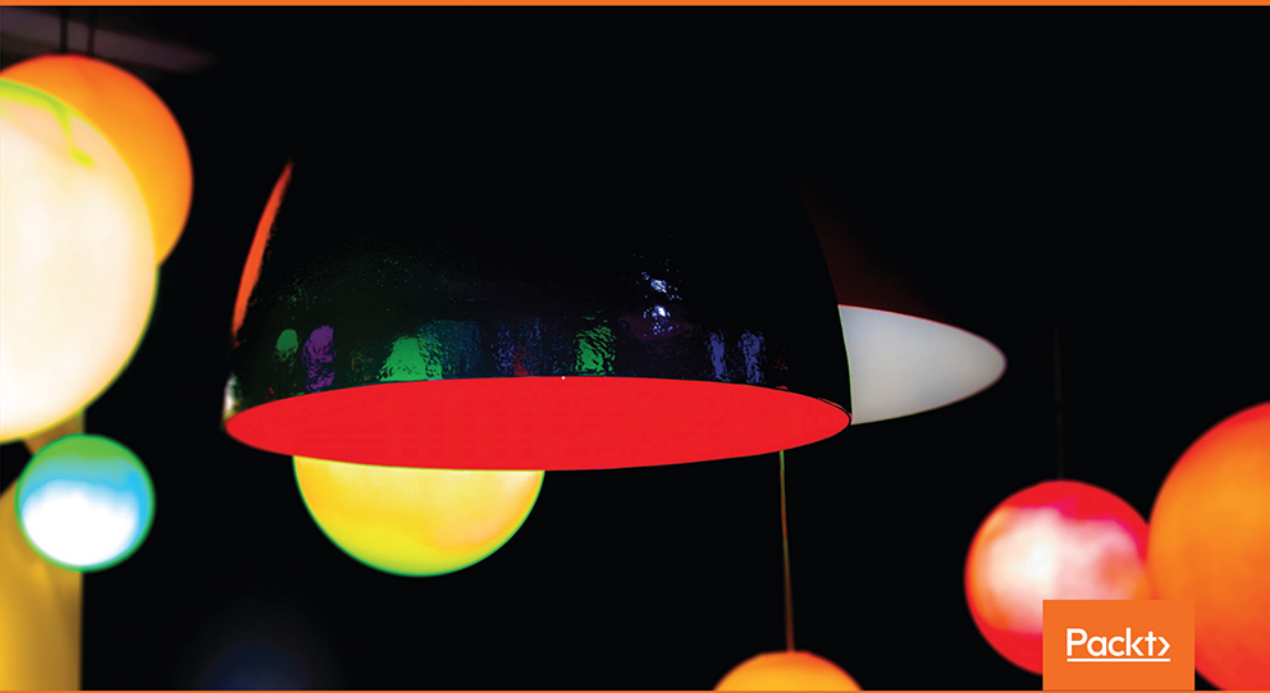


Domain-Driven Design dla .NET Core

Jak rozwiązywać złożone problemy
podczas projektowania architektury aplikacji



Packt 

Alexey Zimarev

Tytuł oryginału: Hands-On Domain-Driven Design with .NET Core: Tackling complexity in the heart of software by putting DDD principles into practice

Tłumaczenie: Lech Lachowski

ISBN: 978-83-283-7765-3

Copyright © Packt Publishing 2019. First published in the English language under the title 'Hands-On Domain-Driven Design with .NET Core – (9781788834094)'.

Polish edition copyright © 2021 by Helion S.A.
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<https://ftp.helion.pl/przyklady/pradd.zip>

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/pradd>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorze	11
O recenzentach	13
Przedmowa	15
Rozdział 1. Dlaczego projektowanie dziedzinowe?	19
Zrozumieć problem	20
Przestrzeń problemu i przestrzeń rozwiązania	20
Co poszło nie tak z wymaganiami?	22
Jak radzić sobie ze złożonością?	23
Rodzaje złożoności	23
Klasyfikowanie złożoności	26
Podejmowanie decyzji i błędy poznawcze	29
Wiedza	32
Wiedza dziedzinowa	32
Unikanie ignorancji	34
Podsumowanie	36
Dalsza lektura	36
Rozdział 2. Język i kontekst	37
Język wszechobecny	37
Język dziedzinowy	38
Wyraźne określanie rzeczy dorozumianych	41
Język i kontekst	47
Podsumowanie	51

Rozdział 3. EventStorming	53
EventStorming	54
Język modelowania	55
Wizualizacja	57
Porady dotyczące organizowania warsztatów EventStorming	58
Kogo zaprosić?	58
Przygotowanie przestrzeni	59
Warsztaty	61
Nasz pierwszy model	68
Podsumowanie	73
Dalsza lektura	74
Rozdział 4. Projektowanie modelu	75
Model dziedziny	75
Co reprezentuje model?	76
Anemiczny model dziedziny	77
Co należy uwzględnić w modelu dziedziny?	78
EventStorming na poziomie projektu	82
Poszerzanie wiedzy	82
Przygotowanie do warsztatów	83
Notacja rozszerzona	83
Modelowanie dziedziny referencyjnej	88
Podsumowanie	91
Dalsza lektura	92
Rozdział 5. Implementacja modelu	93
Wymagania techniczne	93
Rozpoczęcie implementacji	94
Tworzenie projektów	94
Framework	95
Przełożenie modelu na kod	96
Encje	97
Dodawanie zachowania	100
Zapewnianie poprawności	101
Ograniczenia dla wartości wejściowych	102
Obiekty wartości	103
Zdarzenia dziedziny w kodzie	127
Zdarzenia dziedziny jako obiekty	128
Generowanie zdarzeń	131
Zdarzenia zmieniają stan	134
Podsumowanie	139
Rozdział 6. Działanie za pomocą poleceń	141
Wymagania techniczne	141
Otoczenie modelu dziedziny	142
Udostępnianie internetowego interfejsu API	142
Warstwa aplikacji	149

Obsługa poleceń	152
Wzorzec procedury obsługi poleceń	152
Usługa aplikacji	156
Podsumowanie	165
Rozdział 7. Granica spójności	167
Wymagania techniczne	167
Spójność modelu dziedziny	168
Granice transakcji	168
Wzorzec Agregat	173
Ochrona niezmienników	181
Analiza ograniczeń dla polecenia	182
Egzekwowanie reguł	189
Encje wewnątrz agregatu	192
Podsumowanie	201
Rozdział 8. Utrwalanie agregatów	203
Wymagania techniczne	203
Utrwalanie agregatów	204
Repozytorium i jednostki pracy	204
Implementacja dla RavenDB	206
Implementacja Entity Framework Core	221
Podsumowanie	232
Rozdział 9. CQRS — strona odczytu	233
Wymagania techniczne	234
Dodawanie profili użytkowników	234
Zagadnienia dotyczące dziedziny profilu użytkownika	235
Strona aplikacji dla profilu użytkownika	242
Strona zapytań	250
CQRS i konflikt między odczytami a zapisami	250
Zapytania i modele odczytu	252
Implementowanie zapytań	254
API zapytań	254
Implementacja zapytań dla RavenDB	257
Implementacja zapytań dla frameworku encji	264
Podsumowanie	270
Rozdział 10. Event Sourcing	271
Wymagania techniczne	272
Dlaczego Event Sourcing	273
Problemy z utrwalaniem stanu	273
Czym jest Event Sourcing?	277
Event Sourcing wokół nas	278

Agregaty pozyskiwane ze zdarzeń	279
Strumienie zdarzeń	279
Magazyny zdarzeń	281
Utrwalanie oparte na zdarzeniach	283
Podsumowanie	299
Dalsza lektura	300
Rozdział 11. Rzutowania i kwerendy	303
Zdarzenia i zapytania	304
Budowanie modeli odczytu na podstawie zdarzeń	305
Rzutowania	306
Subskrypcje	308
Implementacja rzutowań	311
Subskrypcje nadganiające	311
Rzutowania międzyagregatowe	320
Rzutowanie zdarzeń z dwóch agregatów	321
Wzbogacanie modeli odczytu	331
Kwerendowanie w rzutowaniach	332
Konwertowanie zdarzeń	335
Magazyn trwały	340
Punkty kontrolne	340
Utrwalanie modeli odczytu	344
Zakończenie	349
Podsumowanie	353
Rozdział 12. Kontekst ograniczony	355
Pułapka pojedynczego modelu	356
Zaczynaj od rzeczy małych	356
Znowu złożoność	357
Wielka kula błota	359
Konstruowanie systemów	363
Granice językowe	364
Autonomia zespołu	369
Podsumowanie	373
Rozdział 13. Podział systemu	375
Kiedy, co i jak?	375
Kiedy dzielić?	376
Co dzielić?	378
Jak podzielić?	380
Różne spojrzenia na projektowanie systemów	382
Proces projektowania	383
Dzielenie encji	387
Budowa systemu modułowego	387
Moduły wewnątrz jednego rozwiązania	389
Struktura modułu	392

Budowanie systemu	403
Podłączanie modułów do aplikacji	404
Integracja	405
Interfejs użytkownika	412
Interfejs użytkownika ponad granicami	413
Mikrouслуги	417
Kiedy nie stosować systemów rozproszonych	418
Jak stosować systemy rozproszone	419
Podsumowanie	422
Dalsza lektura	423

EventStorming

Ustalanie terminologii dziedzinowej ma zasadnicze znaczenie, a terminologia ta staje się częścią języka wszechobecnego. Sam proces ustalania terminologii może być jednak dość długi i nie zawsze kończy się pomyślnie. Kiedy omawiamy sposób działania danej firmy oraz problemy, jakie mamy rozwiązać przez napisanie odpowiedniego oprogramowania, zbyt często sprowadza się to do omówienia jedynie funkcjonalności, które firma chce zaimplementować. Zestaw funkcjonalności można oczywiście nazwać *oprogramowaniem*, ale niekoniecznie tworzy on system. Ponadto, aby zbudować kompleksowe rozwiązanie dla konkretnego problemu, trzeba myśleć raczej na poziomie systemu.

W tej książce tylko pokrótce omówiłem myślenie w kategoriach systemów. Aby zgłębić ten temat, możesz zapoznać się z takimi klasycznymi pozycjami, jak *Myślenie systemowe* Geralda Weinberga oraz *Myślenie systemowe. Wprowadzenie* Donelli H. Meadows.

Rodzi się jednak pytanie: Kto nam powie, jak dana firma działa jako system? Z kim powinniśmy porozmawiać i jaką formę powinna mieć ta rozmowa? W tym rozdziale omówię tego typu zagadnienia i spróbuję znaleźć dla nich odpowiedzi.

W tym rozdziale omówione zostaną m.in. następujące tematy:

- metoda EventStorming,
- praktyczne aspekty warsztatów EventStorming,
- poprowadzenie warsztatów,
- co dalej po zakończeniu warsztatów.

EventStorming

W poprzednich rozdziałach dowiedziałeś się, jak ważne jest zrozumienie rzeczywistego problemu. Omówiłem w nich także koncepcję języka wszechobecnego i wyjaśniłem, że nie jest to tylko słownik terminów, ale także opisane słowami zachowania systemu.

Niejasna pozostaje jednak kwestia, jak rozpocząć gromadzenie wiedzy oraz zgłębić komunikację z ekspertami dziedzinowymi, aby lepiej zrozumieć przestrzeń problemu i uzyskać odpowiedni przegląd tego, co ma zostać zbudowane.

Bardzo często można spotkać się z tym, że programiści poznają dziedzinę w postaci wymagań. Omówiłem już ten temat i na pewno zdajesz sobie sprawę, że wymagania mają swoje wady. Wiedzę powinieneś więc poszerzać w bezpośrednich rozmowach z ekspertami dziedzinowymi i na organizowanych z nimi warsztatach lub meetingach. Z reguły wygląda to tak, że przychodzą jakieś osoby z danej firmy i rozmawiasz z nimi przez 2 lub 3 godziny. Omawianych jest wiele rzeczy, pojawia się wiele nowych informacji, ale rezultat w postaci zdefiniowania jakiegokolwiek artefaktu modelowania jest minimalny. Mógłbyś oczywiście zacząć rysować diagramy UML, ale który przedsiębiorca je rozumie? Mógłbyś też robić notatki jedynie po to, aby się przekonać, że potrzebujesz tury lub dwóch warsztatów objaśniających, ponieważ istnieje zbyt wiele mało precyzyjnych i dorozumianych koncepcji, które mają stanowić fundamenty przyszłego systemu, a przez to jest on bardzo trudny do zrozumienia.

Musisz więc rozwiązać kilka podstawowych problemów. Oto one.

- Zapewnij przejrzystość podczas dyskusji. To powinno sprawić, że uczestnicy nie będą przyjmować założeń, gdy wiele osób będzie omawiać to samo za pomocą różnych terminów. Wyeleminuje to również część niejednoznaczności i ujawni pozostałe, aby można je było zbadać.
- Zaproponuj zrozumiały dla obu stron język modelowania. UML nie wchodzi w grę, a zwykłe ramki i strzałki nie zapewniają prawdziwej notacji, więc uczestnicy meetingu mogą poczuć się zdezorientowani i zacząć zadawać pytania mające na celu wyjaśnienie znaczenia różnych kwestii.
- Angażuj wiele osób jednocześnie. Na tradycyjnych meetingach tylko jedna osoba może skutecznie przekazywać komunikaty, podczas gdy wszyscy inni muszą siedzieć cicho i słuchać. Gdy tylko w tym samym czasie zacznie mówić więcej uczestników, rozmowa przestanie być możliwa. Zakładając jednak, że na danej sesji obecne będą osoby zajmujące się różnymi tematami i mające odmienne doświadczenie zawodowe, niektórzy mogą zacząć okazywać brak zainteresowania i się nudzić.
- Znajdź sposób na wyrażenie terminów, zachowań, procesów modelu i decyzji, a nie funkcjonalności i danych.

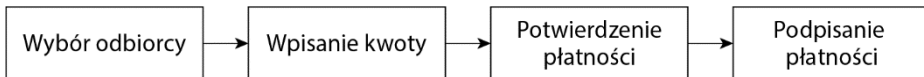
W 2013 r. Alberto Brandolini sformułował metodę, którą nazwał **EventStorming** (czyli burzą zdarzeń). Jej celem było rozwiązanie tych problemów. Omówię ją w tym rozdziale.

Język modelowania

Podstawowa koncepcja techniki EventStorming polega na zapewnieniu prostej notacji modelującej, używanej do wizualizacji zachowań systemu w sposób zrozumiały dla każdego. Takie podejście zapewnia przejrzystość, zwiększa zaangażowanie i wciąga w dyskusję osoby, które w przeciwnym razie w ogóle obawiałyby się udziału w sesji modelowania lub pisania czegośkolwiek na tablicy, jeśli już byłyby obecne.

Jeżeli traktujemy zachowanie jako centralny aspekt wiedzy dziedzinowej, całe ćwiczenie EventStorming ma na celu poznanie sposobu działania danej firmy. Zasadniczo możemy postulować, żeby każdy system w dowolnym momencie znajdował się w określonym stanie. Ten stan mógłby się zmienić, gdyby aktorzy wchodzący w interakcję z systemem wykonywali jakieś działania. Działania tych aktorów powodowałyby zmianę stanu systemu, więc wiedzielibyśmy, że coś się wydarzyło i musimy poradzić sobie z nową sytuacją.

Przyjrzyj się prostemu przykładowi płacenia rachunku z wykorzystaniem bankowości internetowej, który pokazałem na rysunku 3.1.



Rysunek 3.1. Uproszczona sekwencja zdarzeń dla przetwarzania płatności

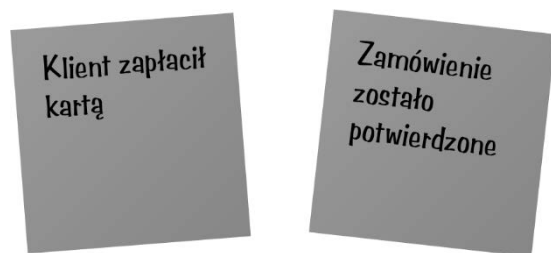
Jak widać na rysunku 3.1, z punktu widzenia osoby dokonującej płatności zmniejszyła się ilość pieniędzy na jej koncie, płatność została zakończona, a rachunek może ona uznać za zapłacony i zapomnieć o nim. Jednak z punktu widzenia odbiorcy rachunek będzie można uznać za zapłacony, gdy otrzyma on pieniądze i będzie mógł dopasować tę płatność do rachunku przy użyciu numeru faktury lub jakiejś innej informacji umieszczonej na rachunku i płatności.

Każde działanie wykonywane przez aktorów w danych systemach spowodowało zmianę stanu. Polecenie zapłaty zostało utworzone i podpisane. Określona kwota została potrącona z konta płatnika, a następnie dodana do konta odbiorcy. Rachunek został oznaczony jako zapłacony. Wszystkie te operacje stały się **faktami**, więc jeśli nie mamy wehikułu czasu, nie możemy ich odwrócić. Gdy odbiorca dowie się, że rachunek został już zapłacony, nie będzie mógł po prostu wszystkiego cofnąć. Będzie mógł jedynie odesłać pieniądze przez zainicjowanie nowej płatności.

Fakty te są zwane **zdarzeniami dziedzinowymi** (ang. *domain events*). Jest to najbardziej podstawowa i zarazem najważniejsza koncepcja metody EventStorming, od której wzięła ona swoją nazwę. Idea zdarzeń dziedzinowych nie jest nikomu obca. Fakty są czymś, co ludzie mogą szybko zrozumieć. Określają coś, co się wydarzyło, a nie zamiar zrobienia czegoś. Nie są funkcjonalnością, formularzem ani przyciskiem. Każde zdarzenie dziedzinowe reprezentuje fakt — zmianę w systemie, który próbujemy modelować.

Dlatego pierwsza część opracowywania języka modelowania polega na utworzeniu koncepcji zdarzeń dziedzinowych. W technice EventStorming każda koncepcja jest reprezentowana przez samoprzylepną karteczkę w określonym kolorze. Użycie kolorów jest niezbędne, ponieważ w miarę wprowadzania do modelu kolejnych elementów kolory pozwalają konsekwentnie reprezentować te same koncepcje w całym modelu, aby uniknąć nieporozumień.

Oryginalną sugestią Alberto jest użycie do reprezentowania zdarzeń dziedzinowych karteczek samoprzylepnych w kolorze pomarańczowym. Najprostszy możliwy model mógłby wyglądać tak, jak pokazałem na rysunku 3.2.



Rysunek 3.2. Zaczynij od czegoś prostego i rozwijaj model

Na rysunku 3.2 pokazałem dwa zdarzenia dziedzinowe, które wystąpiły w sekwencji — najpierw klient zapłacił kartą kredytową lub debetową, a następnie jego zamówienie zostało potwierdzone. Możemy określić to jako dziedzinę handlu elektronicznego (e-commerce). Zdania zapisane na karteczkach samoprzylepnych nie charakteryzują się niczym wyjątkowym oprócz tego, że należy stosować do nich jedną kluczową zasadę: zdarzenia muszą mieć podmiot (rzeczownik) i orzeczenie (czasownik). Czasownik musi być w czasie przeszłym, aby wskazywać, że coś się wydarzyło i stało się faktem.

Wróćmy do przykładu opłacania rachunku. Możemy spróbować określić, jakie zdarzenia powinny się tam znaleźć, co pokazałem na rysunku 3.3.



Rysunek 3.3. Zdarzenia są umieszczane na osi czasu

Na rysunku 3.3 od razu możesz zauważyć kilka rzeczy. Po pierwsze, zdarzenia dziedzinowe są zgodne z osią czasu. Jest to całkiem logiczne, ponieważ fakty przedstawiają kolejne zmiany w systemie, a zatem zdarzają się w określonej kolejności. Płatność nie zostanie np. zatwierdzona przed podpisaniem. Pewne rzeczy mogą dziać się równolegle, np. jednoczesne obciążanie i zasilanie kont, gdy tylko zlecenie płatności zostanie zatwierdzone, co może oznaczać, że bank ma pewność, że płatnik ma wystarczającą ilość środków, aby zrealizować daną płatność.

Po drugie, nie mamy tutaj tylko jednego systemu. Rzeczywiście modelujemy cały proces, ale są co najmniej trzy elementy, które możemy wyraźnie rozróżnić: interfejs użytkownika banku internetowego, za pomocą którego tworzone i podpisywane są zlecenia płatności, zaplecze organizacyjne bankowości, które dokonuje transakcji, oraz własny system dopasowywania płatności do rachunku odbiorcy (nawiasem mówiąc, może on być całkowicie ręczny).

Wizualizacja

Jak widać, nasz prosty model już teraz zapewnia dużą wartość osobom zaangażowanym w warsztaty. Nie tylko spróbowaliśmy określić, co dzieje się podczas procesu płacenia rachunku, ale umieściliśmy również cały przepływ na osi czasu i udało nam się z grubsza zidentyfikować te części procesu, które mogą być realizowane w różnych fizycznych systemach.

Wizualizacja jest jednym z najpotężniejszych aspektów każdej techniki modelowania, a metoda EventStorming nie jest wyjątkiem. Gdy tylko umieścimy coś w swoim modelu, możemy to przeanalizować, zamiast po prostu rzucać pewne słowa i wymachiwać rękami.

Kiedy uczestnicy warsztatów zobaczą cały obraz, niektórzy z nich mogą zacząć zadawać pytania typu „co się stanie, jeśli”. *Co się stanie, jeśli* na koncie nie będzie wystarczającej ilości pieniędzy? *Co się stanie, jeśli* numer referencyjny rachunku będzie niepoprawny? *Co się stanie, jeśli* konto odbiorcy zostanie błędnie podane? I tak dalej. Okazuje się wtedy, że nasz prosty proces ostatecznie nie jest taki prosty, jak mogłoby się wydawać. Pamiętajasz heurystykę dostępności „istnieje tylko to, co widzisz” (ang. *What You See Is All There Is* — WYSIATI)? Nasze wstępne rozumienie opieramy na uproszczonym spojrzeniu na świat. Wszystko działa tak, jak powinno. Nie ma wyjątków ani skrajnych przypadków, ludzie zachowują się odpowiednio i nie zamierzają popełniać błędów, zamierzenie lub nie. Może Cię to zaskoczy, ale prawdziwy świat jest nieco bardziej skomplikowany. Najczęściej liczba przypadków skrajnych przekracza liczbę przypadków uważanych za regularny przepływ zdarzeń. Wszystkie te skrajne przypadki i potencjalne wyjątki stają się znacznie bardziej widoczne, gdy wizualizujemy elementy i umożliwiamy wszystkim ich obserwację.

Mamy tu do czynienia z pewnym problemem, który może być utrudnieniem dla osób starających się utworzyć odpowiedni model zdarzeń. Możesz sobie wyobrazić, że taki warsztat odbywa się w sali konferencyjnej. Zwykle ludzie siedzą wokół stołu i rozmawiają. Jak już wyjaśniałem, nie tak działa metoda EventStorming. Oczekujemy, że uczestnicy będą się poruszać po pokoju i aktywnie angażować w rozmowy, które mogą odbywać się jednocześnie w różnych częściach pomieszczenia. Potrzebujemy więc trochę miejsca. Ale to nie wszystko, czego potrzebujemy. Przyjrzyj się dokładnie prostemu modelowi procesu przedstawionemu na rysunku 3.3. Chociaż wszyscy moglibyśmy się zgodzić, że właśnie wymodelowaliśmy „szczęśliwą” ścieżkę, która nie uwzględnia żadnych skrajnych przypadków ani wyjątków, w prawdziwym życiu ten proces jest o wiele bardziej skomplikowany. Diagram z rysunku 3.3. i tak zajmuje już trochę miejsca w poziomie, a teraz wyobraź sobie modelowanie w ten sposób rzeczywistych scenariuszy. Tradycyjna tablica o długości 2 lub 3 metrów stanowiłaby dla Ciebie pewne utrudnienie.

Wyobraź sobie, że Twój model wygląda tak, jak pokazałem na rysunku 3.4.

Jak widzisz na rysunku 3.4, tablica znajduje się pośrodku. Jednak model nie jest na tyle mały. Jak mawia Alberto, *mój problem jest większy!*

Co się stanie, gdy na tablicy zabraknie miejsca? Pozostała przestrzeń zaczyna być traktowana jak święty zasób. Staje się cenna i zaczynamy oszczędzać miejsce. Niektóre zdarzenia stają się **nieważne** i dlatego nie są umieszczane na tablicy. Część pomysłów staje się drugorzędna i niewarta analizowania. Ostatecznie z powodu konieczności oszczędzania miejsca na tablicy cierpi dyskusja dotycząca modelowania.



Rysunek 3.4. Wizualizacja systemu o rozsądnej złożoności wymaga więcej miejsca

Jest to normalne i tak działa nasz umysł. Jeśli dostrzegamy jakieś ograniczenie, zaczynamy odczuwać jego fizyczną obecność i odpowiednio dostosowywać swoje działania, bez względu na to, jak głupio lub sztucznie będzie to wyglądać z perspektywy czasu. Gdy masz ograniczoną przestrzeń do modelowania, przygotuj się, że model również będzie ograniczony. Zwróć więc uwagę na tę kwestię i postaraj się zapewniać uczestnikom sesji, a zwłaszcza warsztatów EventStorming, jak najwięcej miejsca do modelowania. Bardziej szczegółowe porady podam w następnym podrozdziale.

Porady dotyczące organizowania warsztatów EventStorming

W tym podrozdziale podam kilka praktycznych porad i podzielę się życiowymi doświadczeniami, które ułatwią Ci organizowanie i przeprowadzanie warsztatów EventStorming. Pamiętaj, że zanim zaczniesz przeprowadzać warsztaty EventStorming, nie musisz sprzedawać w swojej organizacji idei projektowania dziedzinowego. Ta dość prosta, ale bardzo skuteczna technika może być przydatna, nawet jeśli nie stosujesz DDD w swoim projekcie. Pomoże Ci w zrozumieniu dziedziny, dla której planujesz pisać oprogramowanie. Tworzy również lepsze relacje między programistami a ich potencjalnymi użytkownikami, ponieważ będą otwarcie omawiać problemy użytkowników i wczuwać się w nie, jednocześnie szukając rozwiązań.

Kogo zaprosić?

Na warsztatach zawsze będziesz potrzebować dwóch rodzajów uczestników — osób z pytaniami i osób z odpowiedziami.

Osoby z pytaniami to programiści i architekci. Zaskakujące jest to, że programiści rzadko biorą udział w sesjach obejmujących bezpośrednio potencjalnych użytkowników oprogramowania, które planują opracować. Poruszyłem ten temat, kiedy w rozdziale 1., „Dlaczego projektowanie dziedzinowe?”, omawiałem oddzielenie przestrzeni problemu i przestrzeni rozwiązań. Powtórzę więc jeszcze raz, że obecność programistów na tych warsztatach jest niezbędna.

Druga grupa osób, po których można by się spodziewać odpowiedzi, a które w rzeczywistości mają więcej pytań, to analitycy biznesowi czy analitycy wymagań — jakkolwiek są oni nazywani w Twojej organizacji. Wprawdzie spędzają dość sporo czasu z potencjalnymi użytkownikami i klientami, ale zazwyczaj podczas warsztatów EventStorming uzyskują nowe informacje, ponieważ jest to spotkanie indywidualne, lecz grupowe.

Ta grupa musi przestudiować dostępne już informacje (ogólne zrozumienie tematu, ewentualnie wymagania lub specyfikacje, jeśli zostały już przygotowane) na temat danej dziedziny i opracować pytania.

Osoby z odpowiedziami to zwykle ci, których nazywamy **ekspertami dziedzinowymi**. Pamiętaj jednak, że nie znają wszystkich szczegółów, a ponadto mogą mieć złudzenie wiedzy. Dlatego na warsztatach potrzebujesz ich jak najwięcej, zwłaszcza że zazwyczaj nie jest łatwo zebrać te osoby w jednym pomieszczeniu ze względu na ich napięty harmonogram. Pośrednio sugeruje to, że znajdują się one generalnie na dość wysokim szczeblu w hierarchii organizacji, ale niekoniecznie tak jest. Musisz dążyć do pozyskania jak największej liczby osób z każdego działu. Powinieneś szukać tych, którzy wiedzą, jak wykonywane są pewne operacje, a nie tych, którzy wiedzą jedynie, jak należy coś zrobić, opierając swoją wiedzę na fikcyjnych opisach i standardach.

Zarówno grupy *osób z pytaniami*, jak i grupy *osób z odpowiedziami* będą musiały przygotować się do warsztatów, starając się jak najlepiej sformułować swoje pytania. Podczas warsztatów pomocne może być zastanowienie się nad kwestiami wymagającymi rozwiązania i bardziej szczegółowe określenie tych kwestii. Część przygotowawcza dotyczy nie tylko programistów. W końcu to dana firma potrzebuje rozwiązania swoich problemów, więc byłoby dobrze, gdyby jej pracownicy byli świadomi własnych potrzeb i dysponowali wystarczającą ilością materiałów, aby móc opisać te potrzeby grupie techników, która ma je zaspokoić za pomocą kodu.

Przygotowanie przestrzeni

Gdy uczestnicy przybywają na warsztaty, bardzo często okazuje się, że odbywają się one w zwykłej sali konferencyjnej, brakuje tablicy, projektor nie działa, a prowadzący ma w swoim sprzęcie wyjściowe gniazdo wideo niekompatybilne ze sprzętem wideo dostępnym w sali. Takie rzeczy są bardzo frustrujące, ponieważ ich ogarnięcie zajmuje sporo czasu, pozostawiając mniej czasu

na właściwą pracę. Co gorsza, tego typu problemy marnują czas nie tylko jednej osoby. Wszyscy zgromadzeni w danej sali muszą czekać i nie mogą wtedy robić niczego innego. Czasami najlepiej jest odwołać spotkanie, aby nie tracić więcej czasu.

Do takich sytuacji nie powinieneś dopuszczać, a warsztaty EventStorming mają określone wymagania, które należy spełnić z wyprzedzeniem, aby uniknąć takiego zamieszania.

Materiały

Z dużym wyprzedzeniem, gdy ustalisz już datę i godzinę meetingu osób z pytaniami i osób z odpowiedziami, zacznij od razu przygotowywać dokumenty.

Najważniejszą kwestią jest przestrzeń do modelowania. Pamiętaj, że Twój problem jest znacznie większy niż jakakolwiek tablica, którą możesz znaleźć w typowym pokoju konferencyjnym (może z wyjątkiem tej, którą zaprojektowałeś do przeprowadzania sesji EventStorming, ale wątpię, że taką masz). Oznacza to, że jako przestrzeni modelowania będziesz musiał użyć ścian. Karteczki samoprzylepne nie trzymają się dobrze na ścianach, chyba że są to szklane ściany. Ponadto po zakończeniu sesji będziesz musiał zdjąć model ze ściany, aby nie przeszkadzał osobom korzystającym z sali po Tobie.

Dlatego będziesz potrzebować rolki papieru. Do warsztatów EventStorming w stylu partyzanckim możesz użyć zwykłej rolki papieru ze sklepu IKEA, która jest sprzedawana jako przestrzeń do malowania dla dzieci. Jest ona jednak dość wąska i najlepszą opcją jest zakup rolki papieru do plotera. Te rolki są szersze (zwykle mają około metra szerokości), dłuższe i wykonane z wyższej jakości papieru.

Będziesz też musiał wymyślić jakiś sposób na przymocowanie papieru do ściany. Konieczne może być sprawdzenie powierzchni ściany w pomieszczeniu, w którym planujesz przeprowadzić warsztaty, i wypróbowanie różnych metod mocowania. Upewnij się, że na ścianie nie ma żadnych elementów, takich jak obrazy, otwory, drzwi czy okna, które utrudniałyby przytwierdzenie do niej rolki papieru.

Oczywiście potrzebujesz dużo karteczek samoprzylepnych. Mam na myśli naprawdę dużo. Nigdy nie wiadomo, ile zdarzeń dziedzinowych rozpiszą uczestnicy, a ostatnia rzecz, jakiej Ci potrzeba, to zbyt mały zapas karteczek samoprzylepnych, który spowoduje, że uczestnicy będą musieli ponownie rozważyć, jakie zdarzenia mogą zapisać. Karteczki samoprzylepne są tanie, a pomysły kosztowne, więc upewnij się, że masz ich wystarczająco dużo, aby możliwe było zapisanie wszystkich pomysłów. Będziesz potrzebował różnych rozmiarów i kolorów. Kwestię notacji przy użyciu kolorów omówię dalej w tym rozdziale. Minimalny zestaw niezbędnych materiałów pokazałem na rysunku 3.5.



Rysunek 3.5. Materiały, których będziesz potrzebować

Ostatnia część ekwipunku to mazaki niezmywalne. Zbyt często zapomina się o nich, jak o czymś oczywistym, a wtedy zostaje się w sali z grupą wysoko opłacanych, bardzo zajętych ludzi i tylko jednym długopisem. To może być bardzo frustrujące. Kup więc wystarczająco dużo mazaków niezmywalnych, najlepiej czarnych, niezbyt cienkich i niezbyt grubych. Najlepiej usunąć jeden mazak na osobę i kilka w zapasie.

Sala

Teraz omówię miejsce. Nie można przeprowadzić warsztatów EventStorming, gdy ludzie siedzą. Powinni raczej stać i swobodnie poruszać się po całej przestrzeni. Dlatego tradycyjna sala konferencyjna z dużym stołem pośrodku i mnóstwem krzeseł wokół nie zda egzaminu. Najpierw musisz się upewnić, że pomiędzy ścianami, na których planujesz przyczepić rolę papieru, jest wystarczająco dużo miejsca, aby uczestnicy mogli się swobodnie poruszać. Najlepiej usunąć wszystkie krzesła lub przynajmniej przenieść w jedno miejsce, z dala od przestrzeni modelowania.

Powinno być miejsce na rozłożenie wszystkich artykułów papierniczych. Dlatego będziesz potrzebować przynajmniej małego stolika gdzieś w rogu sali. Warto również zapewnić widoczną notację, więc uczestnicy powinni do poszczególnych koncepcji używać różnych kolorów. Używanie różnych kolorów do jednej koncepcji jest bardzo dezorientujące i należy tego unikać.

Sesje EventStorming są zwykle bardzo intensywne i wymagają dużo ruchu, myślenia i mówienia, a czasem nawet kłócenia się. Często jest to zabawne, ale może być wyczerpujące. Jako prowadzący powinieneś więc utrzymywać u uczestników wysoki poziom cukru i odpowiednie nawilżenie gardła. Przygotuj przekąski, napoje i owoce — to pomaga. Uczestnicy doceniają takie traktowanie i czują, że zostali zaproszeni na specjalne wydarzenie.

Warsztaty

Przestrzeń jest już przygotowana i musisz zwołać uczestników do sali. Kiedy zauważysz, że zaczynają przychodzić na sesję, skorzystaj ze wskazówek, które podam w tym punkcie, aby pilnować czasu i zwiększyć produktywność sesji. Podam również kilka wskazówek dotyczących tego, jak obserwować i interpretować ludzkie zachowania oraz jak sprawić, by spotkanie było konstruktywne.

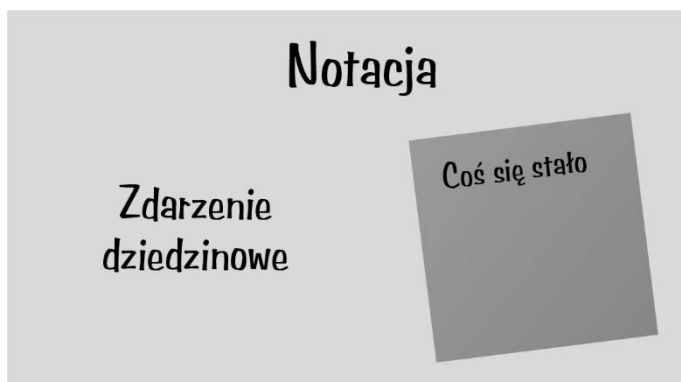
Czas i harmonogram

Zaplanuj co najmniej 2 godziny na jedną sesję. Może wydawać się to niewystarczające (i zwykle nie jest), ale ludziom trudno jest produktywnie pracować przez dłuższy czas w takim środowisku. Przedłużanie warsztatów spowoduje, że dyskusja się wyczerpie, i wywoła poczucie kręcenia się w kółko. Może wynikać to z faktu, że prawie wszystko, co można było omówić, zostało już uwzględnione i umieszczone na ścianie. Dlatego powinieneś oprzeć się pokusie planowania dłuższych warsztatów, takich jak sesje całodniowe.

Pierwsza godzina jest zwykle bardzo intensywna, ale po niej poziom energii spada. Pozwól uczestnikom odpocząć przez 10 minut, poczęstuj ich kawą i owocami, które dla nich przygotowałeś. Często uczestnicy będą dalej rozmawiać w małych grupach o tym, co zostało omówione, więc ich mózgi nadal będą przetwarzać informacje, ale w bardziej zrelaksowany sposób. Po przerwie uczestnicy mają zwykle nowe spostrzeżenia. Będziesz musiał również krok po kroku stosować różne techniki, a przerwa po pierwszej godzinie będzie doskonałą okazją do zrobienia czegoś innego niż to, co robiłeś podczas pierwszej części warsztatów. Omówię te techniki dalej w tym rozdziale.

Początek

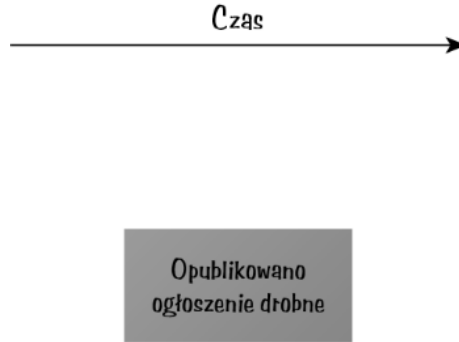
Początek sesji EventStorming z osobami, które nigdy wcześniej nie brały udziału w takich warsztatach, może być dość niezręczny. Jako osoba prowadząca zaczynasz od wyjaśnienia uczestnikom zasad oraz rozdania im mazaków i karteczek samoprzylepnych. Następnie umieszczasz na przygotowanej przestrzeni modelowania pierwszy element notacji, co pokazałem na rysunku 3.6.



Rysunek 3.6. Pierwszy element w notacji

Po tym nastąpi chwila ciszy i uczestnicy zaczną wykonywać niepewne ruchy. Nikt nie będzie wiedział, co robić, a ludzie zwykle nie czują się komfortowo, gdy mają robić coś, czego nie znają. Staje się to szczególnie widoczne, jeśli taka czynność musi zostać wykonana na oczach tłumu.

Taka chwila wymaga od prowadzącego przełamania pierwszych lodów, co wcale nie jest trudne do zrobienia. Organizując taką sesję, wiesz już co nieco o danej organizacji i dziedzinie. Dzięki temu jesteś w stanie wymyślić jakieś zdarzenie dziedzinowe lub nawet kilka. Umieszczenie go na ścianie zapewnia uczestnikom niezbędny przykład, na którym mogą się uczyć. Aktywne branie udziału w sesji nie jest skomplikowane, ale bez przykładu uczestnicy nie czują się pewnie. Dlatego jako prowadzący będziesz musiał przylepić na ścianie pierwszą karteczkę lub kilka karteczek, co pokazałem na rysunku 3.7. Może to być coś istotnego lub celowo możesz napisać na karteczce coś głupawego, próbując wywołać u uczestników śmiech i sarkazm oraz skłonić ich do skorygowania Twojego błędu. Oczywiście z reguły pierwsza reakcja jest taka, że uczestnicy zaczynają mówić głośno o tym, co ich zdaniem powinno zostać umieszczone na ścianie; nie podejmują żadnych działań i oczekują, że Ty zapiszesz wszystkie te rzeczy. W ten sposób odbywają się

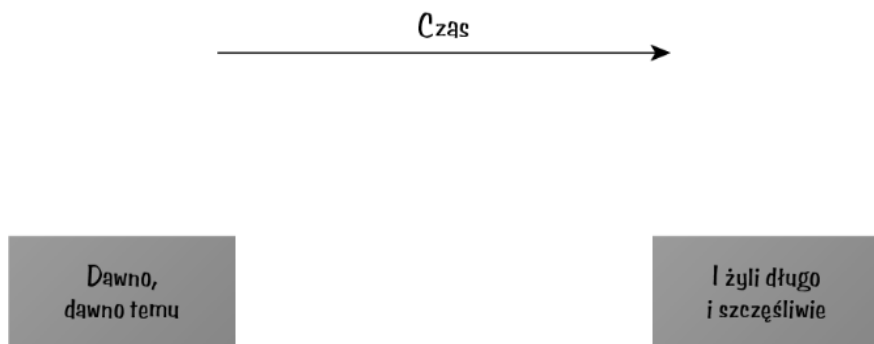


Rysunek 3.7. Losowe zdarzenie umieszczone na środku — przełamywanie lodów

tradycyjne meetingi — zaproszone osoby rozmawiają i można tylko mieć nadzieje, że ktoś będzie robić notatki. Musisz się temu przeciwstawić. Gdy tylko zauważysz, że ktoś wyjaśnia, co musisz napisać na następnej karteczce, lub kwestionuje poprawność tego, co znajduje się już na ścianie, po prostu daj mu mazak i plik karteczek samoprzylepnych. Niech uczestnicy zaczną pisać, tworzyć, dyskutować. Twoim zadaniem ma być obserwowanie i czasem sprowadzenie dyskusji na właściwe tory, na wypadek gdyby uczestnicy utknęli w martwym punkcie.

Znam co najmniej dwie techniki rozpoczynania warsztatów od przyklejenia na ścianie pierwszej karteczki. Autorem jednej z nich jest Alberto Brandolini, który twierdzi, że można umieścić cokolwiek w dowolnym miejscu, byle nie przy lewej krawędzi — nazwijmy ją początkiem przestrzeni modelowania. Należy unikać rozpoczynania od *początku*. Dla ludzi bardzo naturalne jest poszukiwanie struktury i uważamy, że każdy proces gdzieś się zaczyna i kończy. Dlatego logicznie rzecz biorąc, powinniśmy zacząć od początku. Jedynym problemem jest to, że początku nie ma. Po pierwsze, zawsze poświęcamy dużo czasu i energii na dyskusję o tym, gdzie zaczyna się proces, i niczego przy tym nie tworzymy. Po drugie, jest pewne, że przed każdym zidentyfikowanym początkiem będzie coś jeszcze. Dlatego powinniśmy przyklejać pierwszą karteczkę gdzieś na środku i od tego miejsca rozwijać model. To, co nastąpiło przed tym zdarzeniem, zostanie umieszczone po lewej stronie, a to, co ma się wydarzyć później, po prawej.

O innej technice wspomniał Dan North w swoim wystąpieniu na konferencji DDD Exchange 2016. Polega ona na tym, że na jednej karteczce piszemy *Dawno, dawno temu* i przyklejamy ją blisko lewej strony rolki papieru, ale nie na samym brzegu. Na drugiej karteczce piszemy *I żyli długo i szczęśliwie* i przyklejamy ją blisko prawej strony rolki papieru, ale znowu nie na samym brzegu (zobacz rysunek 3.8). To wolne miejsce po obu stronach jest potrzebne dlatego, że jak już wspominałem, na pewno będzie coś przed *Dawno, dawno temu* oraz po *I żyli długo i szczęśliwie*, i będziesz musiał to tam umieścić. Jak widać, aby zapisać te dwie karteczki, nie musisz mieć zupełnie żadnej wiedzy o danej dziedzinie, i jest to dość skuteczna metoda. Uczestnicy zyskają poczucie czasu, a te dwie karteczki na ścianie z wolną przestrzenią między nimi, którą trzeba wypełnić, pobudzą ich wyobraźnię.



Rysunek 3.8. Teraz uczestnicy warsztatów powinni wypełnić całą przestrzeń między tymi dwiema karteczkami

Przebieg warsztatów

Twoja rola prowadzącego nie polega na rządzeniu uczestnikami, ale na ich obserwowaniu i dawaniu wskazówek. Im mniej reguł ustalisz i będziesz egzekwować, tym lepsze będą Twoje warsztaty. Po przełamaniu pierwszych lodów jedni uczestnicy zaczną umieszczać karteczki na ścianie, a drudzy będą zadawać pytania. Na tym etapie powinieneś pamiętać o kilku kwestiach.

- Uczestnicy mają tendencję do zadawania pytań prowadzącemu, ponieważ postrzegają go jako organizatora meetingu, a zatem jako osobę, która ma więcej informacji oraz autorytet. Jako prowadzący kieruj ich pytania do innych osób na sali, a szczególnie do tych zaproszonych osób, które znają odpowiedzi.
- Może dojść do pewnego zamieszania związanego z kwestią zdarzeń dziedzinowych, zwłaszcza jeśli język angielski nie będzie ojczystym językiem uczestników meetingu, a termin *zdarzenie dziedzinowe* będzie przez nich postrzegany jako techniczny (a taki nie jest). W rezultacie mogą znaleźć się osoby, które będą uparcie przyklejać karteczki z pożądanymi przez nie funkcjonalnościami (takimi jak przetwarzanie płatności lub koszyk) lub działaniami imperatywnymi (np. przetwórz płatność lub zarejestruj klienta). Zadaniem prowadzącego jest zapobieganie takim sytuacjom oraz wyjaśnianie, że na tym etapie celem jest opisanie przepływu zdarzeń dziedzinowych, które są faktami i których nie można cofnąć, usunąć ani zmienić.
- Oprócz wspomnianych wyżej kwestii właściwie nie ma więcej błędów lub pomyłek, które mogliby popełniać uczestnicy warsztatów, przynajmniej na poziomie notacji. Nie próbuj deduplikować zdarzeń ani generalizować; nie zniechęcaj uczestników do działania uwagami, że robią coś niewłaściwie.

Z reguły te trzy wymienione wskazówki pomagają poprowadzić warsztaty z organizacyjnego punktu widzenia. Ponieważ jednak mamy do czynienia z ludźmi, zawsze istnieją pewne aspekty behawioralne i personalne. Ważne jest, abyś jako prowadzący zwracał uwagę na kilka rzeczy, które zwykle mogą się wydarzyć, a czasem nawet interweniować.

Po pierwsze, bądź przygotowany na złożone dyskusje. Jeśli nie jest prowadzona żadna dyskusja, może to oznaczać, że dziedzina jest zbyt prosta albo zaprosiłeś niewłaściwych ludzi. Może też chodzić o coś innego, co zniechęca uczestników do zabierania głosu. Spory są nieuniknione, ponieważ każda osoba na sali ma własny punkt widzenia na dziedzinę. Nawet programiści szybko formułują swoje opinie po przeanalizowaniu wstępnego pomysłu lub przeczytaniu specyfikacji. Najważniejsze, żeby programiści zadawali pytania. Nie powinieneś przyjmować żadnych założeń dotyczących tego, co się będzie działo i jak się będzie działo. Zadaniem prowadzącego jest zachęcanie programistów do aktywnego udziału w warsztatach, ponieważ niektórzy z nich są introwertykami i niezbyt lubią brać udział w otwartych dyskusjach. Ponieważ jednak Twoim celem jest umożliwienie programistom lepszego zrozumienia dziedziny, muszą w tej dyskusji uczestniczyć.

Staraj się zwracać uwagę na przypadki skrajne. Uczestnicy zawsze wolą modelować „szczęśliwą” ścieżkę, gdzie nie ma wyjątków i błędów. Zawsze trzeba pamiętać, że jedno zdarzenie jest konsekwencją drugiego zdarzenia, które nastąpiło w określonych okolicznościach. Oczywiście mogą istnieć pewne proste przepływy, ale nie są one powszechne, zwłaszcza gdy mówimy o biznesie. Zdarzenie *przetworzono płatność* może np. logicznie prowadzić do zdarzeń *opłacono zamówienie*, *wysłano zamówienie* i *dostarczono zamówienie*. Co się jednak stanie, jeżeli płatność się nie powiedzie? Co, jeśli kwota płatności nie pokrywa całości zamówienia (płatność częściowa)? A jeśli towary nie są dostępne w magazynie, chociaż jesteśmy przekonani, że są? Co się stanie, jeżeli paczka zaginie podczas transportu? Wszystkie te rzeczy mogą się wydarzyć. Dla programistów może się to wydawać skomplikowane i zwykle nie wiedzą, jak radzić sobie z takimi sytuacjami, poza rzucaniem wyjątku. Jednak firma często ma procedury umożliwiające obsłużenie większości takich sytuacji, a te procedury można i należy wymodelować.

Po drugie, dyskusje na temat przypadków skrajnych prawie na pewno będą powodować niejasności i wątpliwości, ponieważ nie wszystkie wyjątki okażą się objęte procesami biznesowymi lub osoby, które będziesz mieć do dyspozycji na warsztatach, nie będą radzić sobie z takimi sytuacjami. Jeśli na sali będzie kilku ekspertów dziedzinowych, mogą się ze sobą nie zgadzać i kłócić. W przypadku Twojego krótkiego warsztatu takie okoliczności przynoszą efekt przeciwny do zamierzonego. Dlatego gdy w którymś momencie zauważysz, że toczy się jakaś gorąca dyskusja lub wśród uczestników jest zbyt wiele zdziwionych twarzy i nikt nie jest w stanie wyjaśnić pewnej kwestii, możesz zakwalifikować to jako tzw. punkt zapalny (ang. *hotspot*). W tym celu musisz wprowadzić do notacji jeszcze jeden element. Punkty zapalne są zwykle oznaczane karteczkami samoprzylepnymi w jaskrawych kolorach — Alberto proponuje tu np. jaskrawy róż, ale równie dobrze możesz mieć na ścianie coś takiego, jak pokazałem na rysunku 3.9.

Zidentyfikowanie i scharakteryzowanie punktów zapalnych pozwala odłożyć decyzję i przerwać kłótnię, dzięki czemu grupa może pójść naprzód i nie utknąć w martwym punkcie. Pod koniec warsztatów może się okazać, że ściana jest pełna punktów zapalnych, i jest to całkowicie normalne. Oznacza to, że zaproszone przez Ciebie osoby muszą dojść do jakiegoś porozumienia w sprawie obsługi pewnych sytuacji albo musisz porozmawiać z kimś innym i zebrać brakujące informacje. Punkty zapalne zasługują na to, by poświęcić im szczególną uwagę, ale powinno to nastąpić po zakończeniu warsztatów. Staraj się ucinać bezproduktywne dyskusje i kręcenie się w kółko — przyczep do ściany różową karteczkę i poproś uczestników, aby przeszli do kolejnych kwestii.



Rysunek 3.9. Jasne kolory pomagają zwrócić uwagę

Po trzecie, powinieneś pamiętać, że gdy masz kilku ekspertów dziedzinowych, którzy specjalizują się w różnych częściach rozleglejszej dziedziny, będziesz obserwować, jak łączą się w grupy według funkcjonalnego obszaru specjalizacji i tworzą wyspy lub chmury zdarzeń, które ledwie łączą się z innymi wyspami, utworzonymi przez pozostałe grupy. Jest to bardzo interesujące zjawisko i ważne jest, by je zidentyfikować, ponieważ możesz być świadkiem pierwszego szkicu swojej mapy kontekstów. (Mapy kontekstów omówię dalej w książce). Nie zniechęcaj uczestników do takich zachowań i płyn z nurtem. Zwróć uwagę, jak te wyspy łączą się ze sobą. Zwykle niewielka liczba zdarzeń należy do więcej niż jednej grupy zdarzeń dziedzinowych.

Istnieje ponadto prawdopodobieństwo, że dana firma będzie współpracować z organizacjami i systemami, które są poza Twoją kontrolą. Takie podmioty można nazwać **systemami zewnętrznymi** i należy je uwzględnić w modelu. Do takich systemów zewnętrznych mogą być wysyłane pewne zdarzenia dziedzinowe, a inne zdarzenia mogą być od nich odbierane. Wprowadza to do notacji nowy element, a w kolorystyce Alberto systemy zewnętrzne są wizualizowane za pomocą dużych pastelowo różowych karteczek samoprzylepnych, co pokazałem na rysunku 3.10.

Pamiętaj o zapewnieniu nieograniczonej przestrzeni do modelowania i upewnij się, że miejsca jest na tyle dużo, by uczestnicy nie próbowali go oszczędzać. Jeśli jest to potrzebne, przeorganizuj zdarzenia, aby zrobić więcej miejsca, albo najlepiej przyczep na ścianie więcej papieru. Pamiętaj, że papier jest tani, a wiedza cenna. Nie chcesz tracić informacji, ponieważ ktoś musi oszczędzać miejsce na papierze.

Kiedy uczestnikom skończą się pomysły i warsztaty będą przerywane przez niezręczne chwile ciszy, być może będziesz musiał ponownie wskrzesić ogień propozycją spojrzenia na model z innej perspektywy. Istnieją co najmniej dwa stosunkowo łatwe sposoby na wzbogacenie modelu o coś, czego brakuje. Najpierw poproś uczestników, aby przesłędzili oś czasu wstecz. Bardzo często bywa tak, że coś uznano za nieistotne i nie umieszczono tego na ścianie, ale okazuje się to niezbędne, aby mogło dojść do następnego zdarzenia. Ktoś mógł np. zapomnieć,



Rysunek 3.10. Dostawcą płatności jest system zewnętrzny

że przed wysłaniem zamówienia należy sporządzić listę przewozową. Druga technika polega na określeniu, gdzie firma tworzy wartość. Trywialne — podążaj za pieniędzmi. Programiści mają tendencję do omawiania wymyślnych funkcjonalności, które byłoby miło mieć, i zapominają, że firma, aby wypłacić im pensję, musi najpierw coś zarobić.

Na koniec to, o czym już wcześniej wspomniałem. Pilnuj czasu i rób przynajmniej jedną przerwę na godzinę. Dostosuj się do ustaleń i nie przesadzaj; nie zatrzymuj zaproszonych osób dłużej, niż było zaplanowane. Jeżeli posłuchałeś mojej rady i kupiłeś trochę owoców i napojów, niektórzy mogą nawet zechcieć kontynuować, ale to do nich należy decyzja, czy warsztaty powinny potrwać dłużej, niż zakładano.

Po warsztatach

Kiedy czas dobiegnie końca, najprawdopodobniej będziesz mieć długą rolkę papieru z poprzylepianymi wieloma karteczkami w różnych kolorach. Poświęcono wiele czasu i wysiłku, aby powstała, więc ta rolka papieru jest często postrzegana jako cenny artefakt, którym tak naprawdę nie jest. Możesz oczywiście zachować dowody tego, co zostało omówione i wymodelowane, a zwłaszcza listę punktów zapalnych. Jednak główną korzyścią płynącą z przeprowadzonych warsztatów jest wiedza, którą programiści i inni uczestnicy właśnie zdobyli i zabiorą ze sobą do domu. Poszerzenie wiedzy dziedzinowej i zmniejszenie poziomu ignorancji to najważniejsze, choć niewidoczne, artefakty z sesji EventStorming.

Nie oznacza to, że masz wyrzucić tę rolkę. Osoby, które robią to po raz pierwszy, mogą uznać takie działanie za oznakę braku szacunku. Zachowaj rolkę, zrób jej kilka panoramicznych zdjęć i wyślij je wszystkim uczestnikom. Tak — najprawdopodobniej nigdy więcej nie skorzystasz z tej rolki. O wiele bardziej produktywnie będzie utworzenie w razie potrzeby wszystkich tych zdarzeń od podstaw. Uczestnicy mają już wiedzę, a takie dopracowanie modelu będzie dla nich jeszcze korzystniejsze. Jednak ze względów bezpieczeństwa trzymaj rolkę w odpowiednio wyznaczonym miejscu przez kilka tygodni, jak pokazałem na rysunku 3.11.



Rysunek 3.11. Przechowuj rolki przez pewien czas, mimo że prawdopodobnie już nigdy ich nie użyjesz

Pamiętaj, aby zaplanować kolejne sesje w celu omówienia punktów zapalnych. Zwykle mogą w nich uczestniczyć mniejsze konieczne grupy. Czasami będziesz musiał zaprosić jakieś dodatkowe osoby z powodu pytań, na które nie będziesz mieć odpowiedzi. Być może nie będziesz musiał powtarzać wszystkiego, co do tej pory zostało zrobione, i będziesz mógł skoncentrować się na omawianiu odkrytych problemów. Użycie w takich dyskusjach metody EventStorming jest naprawdę korzystne i wzbogaca model.

W następnym rozdziale omówię również, jak przeprowadzać sesje EventStorming na poziomie projektu, co wymaga nieco bardziej technicznego podejścia i co można wykonywać jedynie na mniejszym obszarze funkcjonalnym dziedziny. Wyniki takich sesji można wprowadzać bezpośrednio do systemu śledzenia problemów (ang. *issue tracking system* — ITS).

Nasz pierwszy model

Spróbujmy teraz poćwiczyć i przeprowadzić wymyśloną sesję EventStorming dla dziedziny naszej przykładowej aplikacji. Niełatwo jest wyobrazić sobie, jak mogłoby się to potoczyć, ponieważ najważniejszym aspektem każdego warsztatu EventStorming są ludzie i ich zachowania. Zdecydowanie nie możemy tego odtworzyć tutaj na piśmie, ale możemy sobie wyobrazić dyskusje, które mogą się toczyć, i opracowany w ich trakcie przepływ zdarzeń.

Opiszę fikcyjną sesję EventStorming, podczas której omówiona zostanie aplikacja ogłoszeń drobnych. Prowadząca — nazwę ją Anna — zaprosiła na warsztaty następujące osoby:

- Jana, właściciela firmy, który jest przekonany, że ten system będzie liderem na rynku dzięki swojej prostocie i unikatowym funkcjonalnościom,
- Marię, projektantkę wrażeń użytkownika (ang. *user experience* — UX), która przeprowadziła badania dotyczące istniejących systemów i rozmawiała z kilkoma potencjalnymi użytkownikami,
- Wiktora i Ewę, którzy są programistami kompletnego stosu technologicznego (ang. *full-stack developer*),
- Jakuba, który zarządza zapleczem administracyjnym — zajmuje się finansami i dba o dobrą kondycję firmy.

Nadszedł czas na zaplanowaną sesję i w sali konferencyjnej zaczynają się zbierać zaproszeni uczestnicy, próbując znaleźć krzesła, ale nigdzie ich nie ma. W sali stoją dwa małe stoliki — na jednym znajdują się owoce i napoje, a na drugim leżą stopy karteczek samoprzylepnych w różnych kolorach oraz mazaki niezmywalne, których wystarczyłoby na dwa razy więcej osób, niż zaproszono. Na dwóch ścianach rozwinięto rolkę papieru o długości około 7 metrów. Są też flipchart i biała tablica. Goście wyglądają na nieco zdziwionych i nadszedł czas, aby wyjaśnić im kilka spraw.

— Witamy na naszych pierwszych, ale miejmy nadzieję, że nie ostatnich warsztatach EventStorming — mówi Anna. — Będziemy badać, w jaki sposób firma chciałaby prowadzić działalność, a pod koniec sesji wszyscy obecni na tej sali powinni mieć wspólne zrozumienie tego, co zamierzamy zrobić.

— W tym celu będziemy opisywać, co dzieje się po naszej stronie, gdy klienci korzystają z naszych usług — kontynuuje Anna. — I będziemy do tego używać karteczek samoprzylepnych. Na początek umieścimy na ścianie stwierdzenia faktów, zwane również zdarzeniami dziedzinowymi. Wyobraźcie sobie przepływ zdarzeń w firmie i dla każdego zdarzenia, które zidentyfikujecie, napiszcie kilka słów w czasie przeszłym na karteczce samoprzylepnej i umieśćcie ją na ścianie.

Anna zapisuje na flipcharcie słowo *Legenda* i przykleja na środku pomarańczową karteczkę z napisem *Coś się stało* — *zdarzenie dziedzinowe*. Następnie rysuje poziomą strzałkę na papierze na ścianie i zapisuje pod spodem słowo *Czas*.

— Ponieważ fakty następują po sobie, tworzą sekwencje, inaczej procesy. Fakty nie dzieją się jednocześnie, ale w kolejności, jeden po drugim. Dlatego postaramy się ułożyć te zdarzenia na osi czasu — wyjaśnia Anna.

Wydaje się, że ludzie w pokoju zrozumieli, a Anna rozdaje wszystkim stopy karteczek samoprzylepnych i mazaki. Mimo to nikt nie wydaje się na razie chętny, aby cokolwiek zrobić, i wszyscy zamiast tego patrzą po sobie, czując się trochę nieswojo i nerwowo, jakby obawiali się wykonać jakiś zły ruch. Wtedy odzywa się Jan:

— Najlepiej chyba zacząć od początku. Jaka będzie pierwsza rzecz, która się zdarzyła? Może kiedy klient zarejestrował się u nas?

Taka niepewność wywołuje dyskusję, która prowadzi donikąd. Nadal na ścianie nie umieszczono ani jednej karteczki samoprzylepnej. Anna to zauważa, pisze na karteczce *Opublikowano ogłoszenie drobne* i przykleja ją gdzieś pośrodku.

Pojedyncza karteczka na ścianie wywołuje dyskusję na temat tego, jak powinien zostać nazwany główny element systemu — czy będzie to *Ogłoszenie drobne*, po prostu *Ogłoszenie* czy jeszcze coś innego. Uczestnicy zaczynają umieszczać na ścianie fakty, które teoretycznie powinny się zdarzyć, zanim ogłoszenie zostanie opublikowane, np. *Utworzono ogłoszenie* i *Zaktualizowano ogłoszenie*. Po krótkiej chwili Maria wyraża sceptycyzm co do tych terminów, ponieważ ogłoszeń się nie aktualizuje. Istnieje wiele niezależnych właściwości ogłoszenia, które zmieniają się w różny sposób. Osobno odbywa się np. przesłanie zdjęcia, a następnie aktualizowany jest tytuł ogłoszenia. Możliwość zmiany kategorii ogłoszenia może być ograniczona, a aktualizacja ceny może wywołać pewne interesujące zachowania, takie jak informowanie osób subskrybujących kanał z ogłoszeniami drobnymi, których cena ostatnio spadła.

W tym samym czasie Jan zaczyna omawiać różne zaawansowane funkcjonalności, takie jak ocenianie sprzedawców i kupujących, podczas gdy uczestnicy przyczepiający zdarzenia po drugiej stronie ściany ostatecznie dochodzą do wniosku, że nie ma kupujących ani sprzedających, tylko *użytkownicy*. Ten termin wydaje się mieć znaczenie, gdy mowa o uwierzytelnianiu i profilach, ale nie jest pomocny w procesach sprzedaży i zakupu.

W tym momencie ściana wygląda tak, jak pokazałem na rysunku 3.12.



Rysunek 3.12. Pierwszy model

Część pracy została wykonana i uczestnicy potrzebują krótkiej przerwy, aby skosztować wszystkie smakołyki, które przygotowała dla nich Anna, i zastanowić się nad tym, co zostało omówione i odkryte.

Po przerwie warsztaty są kontynuowane.

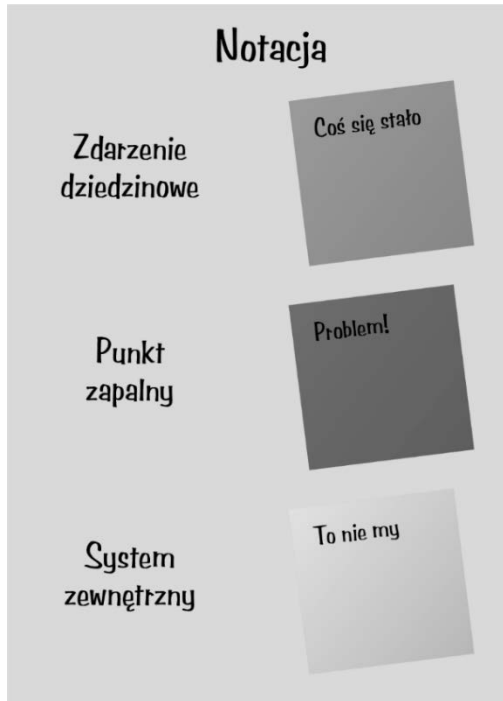
Ewa i Jan zaczynają omawiać proces zatwierdzania, który wcześniej został całkowicie pominięty. Wygląda na to, że na stronach konkurencji umieszczany jest duży odsetek potencjalnie nieuczciwych i ogólnie rzecz biorąc złośliwych ogłoszeń drobnych, a wszystkie te strony stosują pewne mechanizmy zapobiegania takim działaniom. Ci, którzy w ogóle nie zwracają sobie

głowy przeglądaniem publikowanej zawartości, szybko tracą wiarygodność i zaufanie, a w konsekwencji wypadają z biznesu. Jednak nasza firma nie ma personelu, który mógłby nadążyć z ręcznym przeglądaniem wszystkich ogłoszeń, więc dyskusja wciąż kręci się w kółko. Zauważa to Anna i umieszcza na ścianie jasnorożową karteczkę z napisem *Wykrywanie złośliwych ogłoszeń*.

— Wygląda na to, że tego potrzebujemy, ale nie wiemy, jak ma się to odbywać — mówi. — Umieścimy tutaj tę karteczkę jako przypomnienie i wróćmy do tego później, kiedy omówimy resztę.

Dyskusja na ten temat zostaje ucięta i jest konstruktywnie kontynuowana w kierunku ewolucji procesu, który zachodzi przed wykryciem złośliwego ogłoszenia i po.

Na tym etapie do notacji zostaje dodany trzeci element, który pokazałem na rysunku 3.13.



Rysunek 3.13. Finalna notacja dla warsztatów na poziomie ogólnym

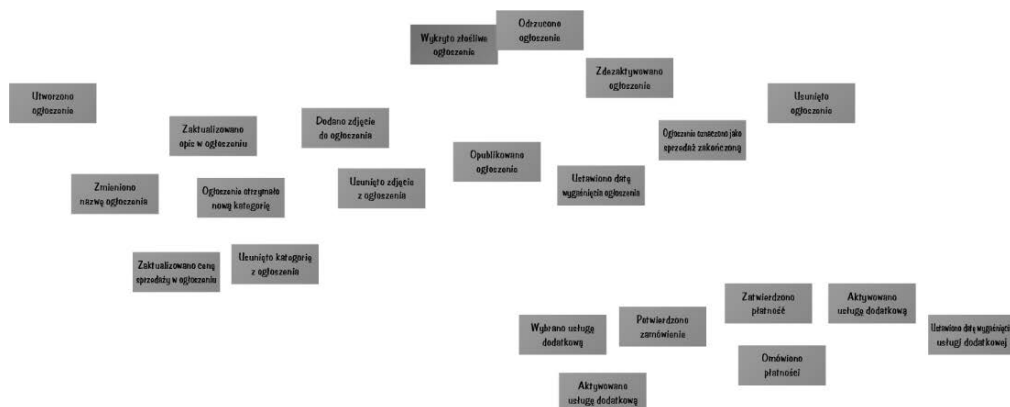
Jakub nie był dotąd zbyt aktywny w dyskusji i tylko przechadzał się po sali, kiwając głową i czasami robiąc grymas sprzeciwu. Kiedy Anna pyta go, co jest nie tak, Jakub rzuca zniecierpliwiony:

— Nikt nawet nie pomyślał o tym, w jaki sposób mamy zarobić jakiegokolwiek pieniądze. Bez zarabiania nie przetrwamy. Nie mamy tak dużych możliwości inwestycyjnych i lepiej, żebyśmy jak najszybciej zaczęli osiągać dochód.

Wywołuje to wśród uczestników pewną konsternację i zaczynają się zastanawiać, jak mogli zapomnieć o uwzględnieniu aspektu zarabiania. Wtedy Jan wyjaśnia pierwotny pomysł, który polega na tym, że podstawowa usługa jest bezpłatna, ale niektóre dodatkowe usługi, takie jak umieszczanie ogłoszeń na początku wyników wyszukiwania czy wyświetlanie większych zdjęć, będą oferowane za niewielką opłatą. Wyjaśnia również, że bezpłatna usługa będzie dostępna tylko dla osób fizycznych, ale gdy firma będzie chciała coś sprzedać za pośrednictwem tej strony internetowej, będzie musiała mieć umowę o wyłączność. Poza tym, kontynuując, sprzedaż samochodów i nieruchomości powinna wyglądać zupełnie inaczej, ponieważ wymaga zaawansowanej integracji i pewnych środków bezpieczeństwa, więc te usługi nigdy nie będą darmowe.

Ta nowa informacja wywołuje różne dyskusje i po krótkiej chwili wszyscy się zgadzają, że w pierwszej wersji muszą dotrzeć do jak największej liczby odbiorców i dążyć do możliwie prostej usługi. Oznacza to, że będą oferować tylko bezpłatne ogłoszenia z kilkoma opcjami, które powinny wygenerować podstawowe źródło przychodów.

Warsztaty dobiegają końca, a ściana jest pokryta karteczkami samoprzylepnymi. Na rysunku 3.14 pokazałem, co udało się opracować uczestnikom.



Rysunek 3.14. Ostateczny model opracowany na warsztatach ogólnych

Jak widać na rysunku 3.14, sesja ujawniła kilka niewyjaśnionych kwestii, które były nieznaną częścią uczestników albo co do których czynili oni pewne założenia. Oto kilka przykładów.

- Nie ma czegoś takiego jak *Zaktualizowano ogłoszenie*. Są raczej takie zdarzenia, jak *Przesłano zdjęcie*, *Obniżono cenę* itd., które są bardziej precyzyjne i wyzwają inną logikę.
- Istnieją różne konteksty. W jednych osoba korzystająca z serwisu jest znana jako *użytkownik*, a jednocześnie w innych jako *sprzedawca* lub *kupujący*. Relacja między tymi kontekstami nie jest stabilna, co może wskazywać, że są to co najmniej różne byty.
- **Minimalny funkcjonalny produkt** (ang. *Minimal Viable Product* — MVP) skurczył się do absolutnego minimum bezpłatnych ogłoszeń z niewielkimi płatnymi dodatkami. Wszystko inne ma zostać opracowane później.

- Funkcjonalności takie, jak ocenianie sprzedawców i kupujących oraz inteligentne podpowiedzi, są przyjemne, ale nie przynoszą natychmiastowej wartości użytkownikom.
- Będzie natomiast duże zapotrzebowanie na system wykrywania złośliwej zawartości, ponieważ ręczne sprawdzanie będzie działać jedynie przez krótki okres, jeśli wziąć pod uwagę liczbę ogłoszeń, które mają być publikowane dziennie.

Programiści znacznie lepiej zrozumieli, czego firma oczekuje od systemu i jak powinien on wyglądać według ekspertki od UX. Dokonali wielu poprawek oryginalnych pomysłów i wydaje się, że wszyscy są na tej samej drodze. Poziom ignorancji jest znacznie niższy niż przedtem i znacznie zmniejszyło się ryzyko, że programiści będą zajęci rozwiązywaniem niewłaściwego problemu.

Podsumowanie

W tym rozdziale dowiedziałeś się, na czym polega metoda EventStorming i dlaczego każdy zespół może skorzystać na zorganizowaniu takich warsztatów z ekspertami dziedzinowymi, czyli na zdobyciu wiedzy dziedzinowej i zmniejszeniu poziomu ignorancji. Podąłem również kilka praktycznych wskazówek dotyczących sposobu organizowania i przeprowadzania sesji EventStorming.

Ostatnią część rozdziału poświęciłem modelowi naszego przykładowego biznesu. Omówiłem przepływ biznesowy, określiłem wiele zdarzeń oraz kilka punktów zapalnych. Dzięki temu wiesz już sporo o tym, jakiego rodzaju system będziesz budować w tej książce.

Pokrótko omówiłem kwestię behawioralnego aspektu sesji EventStorming, ale jest to znacznie szerszy temat, któremu nie jestem w stanie poświęcić więcej miejsca. Na końcu rozdziału znajdziesz odwołania do materiałów, których lektura pomoże Ci dowiedzieć się więcej o ludziach, ich uprzedzeniach i zachowaniach oraz o tym, dlaczego tworzenie oprogramowania jest procesem uczenia się.

W następnym rozdziale opiszę szerzej proces modelowania, koncentrując się bardziej na artefaktach, które mogą pomóc Ci w rozpoczęciu pisania kodu i dostarczaniu wstępnych prototypów tak szybko, jak to możliwe.

Spółeczność EventStorming toczy wiele dyskusji na temat wykorzystania tej techniki w przypadku zespołów rozproszonych lub gdyby nie można było zapewnić bezpośredniego udziału użytkowników końcowych w sesjach. Alberto zwraca uwagę, że mowa ciała jest niezwykle ważna dla ogólnego odczuwania przebiegu sesji oraz określania ról uczestników w sesji i w działaniu biznesu. Zgadzam się z Alberto, ale często mamy do czynienia z sytuacjami, w których zorganizowanie takiej sesji jest prawie niemożliwe. W przypadku systemów rozproszonych polecałbym korzystanie z narzędzi internetowych działających w czasie rzeczywistym, takich jak Miro, którego użyłem do utworzenia wielu diagramów z tej książki. Pozwala to zapewnić poszczególnym osobom uczestnictwo w sesjach modelowania niezależnie od miejsca, w którym się znajdują.

W przypadku biznesów typu oprogramowanie jako usługa (ang. *Software as a Service* — SaaS) problem może być jeszcze trudniejszy, ponieważ nie ma jednej grupy użytkowników, którą można by uznać za reprezentującą całą bazę użytkowników. Jednak nawet w takim scenariuszu często można zidentyfikować grupę najbardziej zaangażowanych klientów i zaprosić ich do swojego zespołu. W ten sposób zyskujesz nie tylko doskonałe informacje o tym, jak korzystają z Twojego systemu i jakich ulepszeń oczekują, ale także świetną darmową reklamę przekazywaną z ust do ust. Ludzie zasadniczo szanują otwartość i bardzo doceniają, gdy ich głos jest słyszany przez programistów.

Dalsza lektura

Oto lista materiałów, które pomogą Ci poszerzyć wiedzę:

- strona <https://eventstorming.com> — znajdziesz na niej więcej informacji na temat metody EventStorming oraz odnośniki do innych materiałów,
- Alberto Brandolini, *Introducing EventStorming*, Leanpub, 2017 (https://leanpub.com/introducing_eventstorming).

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Programowanie dziedzinowe: tak się trafia w sedno potrzeb!

Wdrożenie zasad projektowania dziedzinowego (DDD) jest szansą na uzyskanie doskonałych wyników projektowania oprogramowania dla złożonych wymagań biznesowych. Aby podejście DDD ujawniło swoje zalety, konieczne jest zrozumienie potrzeb użytkowników i zidentyfikowanie właściwych problemów do rozwiązania. Dopiero potem można przystąpić do budowania modeli. Zaangażowanie zainteresowanych stron biznesowych w ten proces nie zawsze jednak przebiega efektywnie. Konieczne jest zrozumienie natury modeli dziedzinowych opartych na zachowaniu oraz tego, jaką rolę odgrywają one w budowie zwinnych i zmodularyzowanych systemów.

Dzięki tej książce zrozumiesz praktyczne znaczenie złożoności dziedziny i behawioralnych aspektów języka dziedzinowego. Poznasz podstawowe zasady DDD i nauczysz się używać takich narzędzi jak EventStorming, Event Sourcing i CQRS. Dowiesz się, jak stosuje się DDD do różnych stylów architektonicznych, takich jak REST, systemy reaktywne i mikrousługi. Nauczysz się pracować we frameworku .NET Core 2.2, aby przetłumaczyć modele dziedzinowe na wykonywalny kod C#, oraz Vue.js, aby zbudować frontend aplikacji. Nie zabrakło tutaj również zagadnień refaktoryzacji kodu, numerowania wersji zdarzeń i migracji. W rezultacie poszczególne zespoły w organizacji będą mogły pracować elastyczniej i efektywniej dzięki ulepszonym usługom i oddzielnym interakcjom.

W tej książce między innymi:

- angażowanie interesariuszy biznesowych do rozwiązywania złożoności dziedziny
- koncepcja kontekstu ograniczonego i agregatu
- projektowanie modeli tymczasowych na podstawie zachowania
- wzorzec CQRS i modele odczytu wykorzystujące rzutowania
- jednokierunkowy przepływ interfejsu użytkownika w Vue.js
- praktyczne stosowanie zasad projektowania dziedzinowego

Alexey Zimarev jest certyfikowanym instruktorem Google z wieloletnim doświadczeniem. Wykłada Pythona, uczenie maszynowe i głębokie, algorytmikę oraz zagadnienia big data. Przez ostatnie lata pracował w rządowym laboratorium Kanady nad projektem z zakresu uczenia maszynowego. Obecnie zajmuje się algorytmami używającymi GPU do optymalnego trenowania złożonych modeli uczenia maszynowego.

Helion 	<i>Sprawdź nasze szkolenia!</i>  SZKOLENIA AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	KOD KORZYŚCI Sięgnij po więcej!   ISBN 978-83-283-7765-3  9 788328 377653
 helion.pl		
 0 801 339900		
 0 601 339900		
INFORMATYKA W NAJLEPSZYM WYDANIU		Cena: 99,00 zł

Packt