

ВСТУПНЕ СЛОВО

Сьогодні вже зрозуміло, що звичайного підходу до бізнесу недостатньо для того, щоб відповісти на глобальні виклики, які стоять перед світом. Якщо ми хочемо йти в ногу з динамікою змін, нам потрібно буде розробляти нові креативні рішення і навчитися масштабувати їх набагато швидше. Саме про це наша книга. Вона розповідає історію компанії Theodo, технологічного піонера, що знаходиться в самому центрі цієї боротьби, розробляючи нові технологічні системи для стартапів і корпорацій. За останнє десятиліття компанія Theodo знайшла нові, більш ефективні способи взаємодіяти як всередині компанії, так і з клієнтами, а також розширювати масштаби своєї діяльності, оскільки штат компанії збільшився з 10 до 700 осіб, а її дохід склав 100 мільйонів доларів США.

Theodo досягла цього, постійно кидаючи виклик найкращим бізнес-практикам, включаючи Agile, і шукаючи нові, досліджуючи, що працює в технологічному бізнесі, а що — ні. Свої уроки компанія виклала в Маніфесті Lean Tech, щоб інші могли вчитися на її досвіді. Уроки, якими вона ділиться, стосуються не лише технологічних фірм, але й сталого бізнесу, який намагається уникнути бюрократичних перепон, що гальмують його зростання.

Ключовим джерелом натхнення для Theodo та ефективною протидією від бюрократії стало Lean мислення та набір цінностей, принципів і практик, що лежать в основі бізнес-системи Toyota, яка вивела компанію на перше місце в автомобільній індустрії. Понад 30 років тому західні автовиробники намагались знайти пояснення факту, що Toyota може виробляти автомобілі з нульовим рівнем дефектів, витрачаючи на це вдвічі менше людських зусиль і коштів. Toyota здійснила цей прорив, розвиваючи навички вирішення проблем у передових командах, щоб вони могли покращити свою роботу для досягнення принципу «правильно з першого разу, вчасно і щоразу» і аналізувати першопричини кожного збою в роботі. Згодом вони зможуть пов'язати попередні та наступні етапи в єдиний потік «точно вчасно» від сировини до готової продукції, що створюється під впливом попиту. Знаменита Виробнича система Toyota (TPS) виявилася навчальною системою для синхронізації та постійного вдосконалення будь-якої організації, незалежно від того, наскільки вона складна.

Менш відомо, що Toyota також навчилася розробляти новий автомобіль за вдвічі менший час і з вдвічі меншими зусиллями та витратами, перетворивши традиційне «каскадне» управління проектами на систему безперервного розвитку для наступних поколінь продуктів. Знову ж таки, команда відстежує прогрес і відразу ж вирішує проблеми, тому кроки можуть бути чітко синхронізовані. Цю систему очолює головний інженер, який збирає інформацію та відгуки від клієнтів, розробників та виробництва і вирішує, що потрібно змінити, а що залишити незмінним для наступного покоління продукту, замість того, щоб змінювати все одразу.

Ця книга також розповідає про те, як ці ідеї мали глибокий вплив на піонерів технологічної індустрії Стіва Джобса та Джеффа Безоса і призвели до появи багатьох нових практик розробки програмного забезпечення в рамках руху Agile, таких як екстремальне програмування, Scrum та DevOps. Особливо важливо, і це, мабуть, є найбільшим її внеском, що вона також описує, як можна розширити масштаби практики Agile, впровадивши їх в організацію, що навчається, зосереджену на розвитку здібностей через систематичне навчання шляхом вирішення проблем.

Ці відмінні способи спільної роботи також ставлять нові виклики перед лідерами, про що автори Фабріс Бернхард і Бенуа Шарль-Лавозель дізналися, створюючи і розширюючи власну систему Lean у компанії Theodo протягом останнього десятиліття. Вони зрозуміли, що природа прийняття рішень змінюється — від придбання експертизи до виконання роботи для вас, до найму потрібних талантів і розвитку внутрішніх можливостей. Напрямок задається шляхом формулювання прогалин у продуктивності, які необхідно ліквідувати, і узгодження діяльності команд навколо них. Нарешті, розвиток здатності команд вирішувати проблеми, щоб закрити ці прогалини, відбувається завдяки залученню найдосвідченіших співробітників, як вчителів і наставників.

Це важлива книга. Вона описує бізнес-систему, якої так відчайдушно потребують організації в усьому світі, і нагадує нам, що коли йдеться про вирішення складних і нагальних проблем сучасності, людське прийняття рішень і командна робота так само важливі, як і технології.

Деніел Т. Джонс
Монмут,
Велика Британія

ПРИМІТКА ДО ОБКЛАДИНКИ

Ієрогліф на обкладинці — це ієрогліф кандзі (японська мова) або ханьцзи (китайська мова), що позначає поняття «людина». Він складається лише з двох штрихів і нагадує людину, яка стоїть або йде.



Це нагадування про те, що в основі як Lean мислення, так і Agile лежить «повага до людини», бачення потенціалу в кожній людині та наполеглива праця над створенням середовища, де люди можуть реалізувати свій потенціал, бути креативними і по-справжньому насолоджуватися своєю роботою.

ПРИМІТКА АВТОРІВ

Щоб надати контекст читачам, яких зацікавила наша історія, ми додали цей розділ, який є необов'язковим для розуміння подальшого змісту книги.

НАША ПОДОРОЖ

Втеча з корпоративного світу

4 березня 2006 року Роджер Федерер захистив свій титул проти Рафаеля Надаля у фіналі тенісного чемпіонату в Дубаї. На той час Федерер встановив своє домінування в тенісі. Бувши першою ракеткою світу протягом двох років, його серія перемог, яка на той час складала 81 з 85 матчів, робила його майже непереможним. Майже, якби не одна висхідна зірка, Рафаель Надаль, який лише за три зустрічі вже двічі переміг Федерера. Для тенісних фанатів ця четверта зустріч у Дубаї стала переломним моментом, який покаже, чи були перемоги Надаля випадковістю, чи початком одного з найзапекліших суперництв в історії тенісу.

Бенуа — суперфанат тенісу, і він не хотів пропустити цю подію. На жаль, для перегляду гри потрібен був доступ до нішевого платного телеканалу, якого у Бенуа не було. Не розгубившись, він вирушив на пошуки місця з відповідною підпискою і можливістю транслювати гру. У великому місті з сотнями барів це не повинно бути надто складно, подумав він.

Блукаючи районом, відомим своєю великою кількістю барів, Бенуа почав хвилюватися. У закладах, які він відвідував, або не було каналу, або не хотіли показувати спортивні трансляції в середині дня, або вже показували іншу гру. Коли гра вже починалася, Бенуа все ще продовжував шукати. Його охопив відчай.

Через дві години після того, як Бенуа вийшов зі своєї квартири, він нарешті пройшов повз *Bombardier*, типово англійський паб, який поділяв його пристрасть до спорту і показував гру на одному зі своїх екранів. З полегшенням Бенуа нарешті зміг сісти і насолодитися грою, але на той час він пропустив перший тайм. Після гри він зателефонував Фабрісу, щоб поділитися своїм розчаруванням.

Ми переконали себе, що цю проблему може і повинен вирішити інтернет-стартап. Майже закінчивши навчання, ми вхопилися за можливість вирватися з корпоративного світу, який чекав на нас, і запустили *Allomatch.com*, який зараз є частиною *Fanzone.com*, сайт для пошуку барів для кожного спортивного вболівальника, який коли-небудь запитував: «Хто-небудь знає місце, де будуть показувати...?»

Фабріс створив першу версію *Allomatch.com* за одну ніч. У той час ми нічого не знали про професійні технології та методології розробки програмного забезпечення. У нас також не було ні грошей, ні часу, оскільки ми були студентами. Ми зосередилися на тому, щоб зробити продукт якомога швидше і дешевше.

Першу версію ми нашвидкуруч зібрали разом, використовуючи *Joomla*, одну з доступних на той час систем управління контентом з відкритим вихідним кодом. Потім ми дуже швидко залучили клієнтів, тому що найефективнішим способом отримати інформацію про те, коли в барі показують гру, було попросити власників бару надавати її щотижня самостійно. Після того, як кілька перших користувачів погодилися надавати інформацію, ми підтримували ці цінні відносини, швидко оновлюючи продукт, щоб показати їм, що ми серйозно ставимося до їхніх відгуків.

Зрештою, ми зробили продукт платним, щоб переконатися, що є клієнти, готові платити за нашу нову послугу, перш ніж ми повністю присвятили себе проєкту. Протягом перших двох років *Allomatch.com* зосереджувався на створенні ефективного надання цінності, що змусило нас орієнтуватися на користувача та дало змогу використовувати ітеративний підхід до продукту, який допоміг нам залучити сотні клієнтів і збільшити трафік до мільйонів відвідувачів на рік.

Наш початковий швидкий ітеративний підхід не завадив нам наштовхнутися на дві перешкоди, одну з технічного боку, а іншу — з бізнес-сторони.

З технічного боку, через два роки нас наздогнав технічний борг. Коли Фабріс почав кодувати *Allomatch*, він був недосвідченим аспірантом. Як наслідок, *Allomatch* не слідував кращим галу-

зевим практикам. Накопичення швидких і «брудних» ітерацій ускладнювало створення кожного нового додатку. Ми ледь не просили вибачення перед новими розробниками, яких наймали.

Ми були не єдиними у світі, хто стикався з цими проблемами в той час. У вебіндустрії поширювалося нововведення, спрямоване на усунення саме цього технологічного вузького місця: фреймворки MVC (модель-вид-контролер) з відкритим кодом. Це були безкоштовні, готові архітектури: Django для Python, Ruby on Rails для Ruby та Symfony для PHP. Вони були доступні будь-якому розробнику, навіть початківцю. Це було схоже на бажання побудувати будинок і отримати від когось міцний фундамент безкоштовно. Для таких інженерів, як ми, було легко зрозуміти цінність такого фреймворку: він прискорював розробку і забезпечував направляючі рейки, які допомогли нам уникнути створення кодової бази «спагеті».

Після чотирьох місяців напруженої роботи кодову базу нарешті було перенесено на Symfony, і це було як небо та земля з технічного боку. Фреймворк виконав свою функцію — спростив впровадження еволюцій та забезпечив кращу якість завдяки вбудованим найкращим практикам. Загалом, Symfony дозволила кодовій базі Allomatch рости ще 12 років без необхідності переписування.

Друга перешкода була з боку бізнесу. Через два роки Allomatch не генерував достатнього доходу, щоб забезпечувати свою діяльність. Наша зосередженість на ефективності дозволила знизити витрати, але ми ледве заробляли достатньо, щоб бути прибутковими. Наш швидкий ітеративний підхід допоміг нам залучити понад сотню клієнтів, але ми не зробили необхідного кроку назад, щоб зрозуміти, що працюємо на занадто маленькому ринку. Незважаючи на те, що ми доклали багато зусиль, ми не змогли знайти спосіб переорієнтувати Allomatch на інший, більший ринок.

Одного пізнього вечора, гуляючи вулицею після роботи, ми говорили про брак варіантів і тривожно низький рівень прибутку. Тоді ми згадали, що за кілька днів до цього наш знайомий підприємець сказав нам, що хоче створити власне програмне забезпечення, яке допоможе йому розширити бізнес. Ми порекомендували його комусь іншому, але потім зрозуміли, що втратили можливість заробити, яка могла б втримати компанію на плаву ще кілька місяців. Ми зателефонували йому і переконали, що можемо допомогти. Разом ми створили технологічний продукт, який змінив правила гри для його бізнесу.

Він допоміг йому розширюватися значно швидше, скоротивши час, необхідний для обслуговування клієнтів, з одного тижня до трьох годин. Коли це помітив великий гравець у своїй галузі, компанія запропонувала придбати його бізнес. Продукт продовжив своє існування в компанії, що його придбала і забезпечував зростання ще протягом десятиліття.

Коли цей перший експеримент зі створення технологічних продуктів для клієнта виявився водночас і захопливим викликом, і величезним успіхом, ми побачили можливість розпочати свою діяльність у новій галузі. Так почалася компанія Theodo: як розробник технологічних продуктів для стартапів та малого бізнесу.

Theodo

Знайти перших клієнтів для Theodo не було особливо складно. Багато стартапів та малих і середніх підприємств потребували створення технологічного продукту, але не мали для цього можливостей і намагалися знайти постачальника, якому можна було б довіряти. Завдяки Allomatch.com як легкодоступному референсу та нашій успішній міграції на фреймворк Symfony, ми могли продемонструвати потенційним клієнтам наш продукт та технічну експертизу.

Чого у нас не було, так це сильної методології. Ми спробували запропонувати лише підхід Agile, який здавався найближчим до нашого досвіду створення Allomatch, але наші клієнти не хотіли відмовлятися від гарантій заздалегідь визначених специфікацій та фіксованої ціни. Тож ми спробували творчо поєднати традиційний проєктний менеджмент з практиками Agile. На жаль, ця суміш поєднала в собі найгірше з обох світів: затягнутий старт через довгу фазу специфікації, команди, які самотійно розбиралися з реалізацією, і непорозуміння, виявлені наприкінці.

Проєкт 2011 року проілюстрував, наскільки поганим може бути такий підхід. Двоє братів звернулися до нас, щоб створити платформу для онлайн-консультування. Вони були захоплені своїм проєктом і були сповнені рішучості зробити щось значуще. Оскільки це був новий продукт, ми вважали, що важливо мати можливість ітерацій, і порадили їм використовувати для цього методи Agile. Але вони наполягали на тому, що точно знають, чого хочуть, тому ми почали працювати над їхніми початковими вимогами, не надто ставлячи їх під сумнів.

Однак у міру того, як ми створювали продукт, їхнє розуміння того, чого вони хотіли, змінювалося. Вони просили внести відповідні зміни та нові функції, і ми неохоче погоджувалися на незадовільні компроміси. Бюджет був заздалегідь визначений, тому кожна зміна означала додаткову незаплановану роботу і витрати. Гірше того, коли вони нарешті почали тестувати ринок, то зрозуміли, що платформа, якою вони її собі уявляли, була абсолютно марною. Їм потрібно було щось інше, і вони вирішили відмовитися від усього цього. Це був великий провал для всіх.

Наприкінці проекту вони запросили нас на чарку, щоб підбити підсумки. Один з братів був дуже відвертим і сказав, що ми були вправні технічно, але зіпсували продукт. Це було важко чути, і ми заперечили йому: він був тим, хто відкинув нашу рекомендацію працювати в повністю гнучкому режимі.

Ми ніколи не забудемо його відповідь: «Коли я йду до лікаря, я не приходжу з рецептом. Ви — експерти, і ви маєте сказати мені, як діяти далі. Якщо Agile — це шлях до успіху, то саме ви маєте нав'язати його нам».

Ця розмова була шоком і відкриттям. Звичайно, він мав рацію. Ми не могли перекладати відповідальність за знання правильної методології на нашого клієнта. Якщо ми не вірили у свій метод роботи, то настав час щось з цим зробити.

Натхненні викликом двох братів, ми вирішили докорінно змінити наш підхід до роботи з наступними клієнтами. Ми вирішили піти ва-банк з Agile і або досягти успіху в тому, щоб зробити наших клієнтів щасливими, або переглянути наші зобов'язання щодо розвитку консалтингової компанії.

Йдемо ва-банк з Agile

Рішення про залучення клієнта та ітерації над продуктом, щоб стати повністю Agile, було не таким простим. Agile не був сумісний з нашою бізнес-моделлю з фіксованою ціною. Якби ми залучили клієнта та майбутніх користувачів під час розробки продукту і дозволили їм просити про зміни на кожній ітерації, це зробило б початкові специфікації та відповідне ціноутворення спірними після перших кількох ітерацій. Наші контракти базувалися на цих початкових специфікаціях і цінах.

Ми вирішили посилити нашу експертизу в Agile та Scrum, пройшовши тренінг з Джеффом Сазерлендом, співзасновником Scrum.