Представьте, что вы обслуживаете комплексное решение, от которого зависит жизненно-важные бизнес-процессы вашей организации (вашей компании).

Представьте, что вам поручили разработать и поддержать комплексное программное решение, от правильной и точной работы которого зависит успех вашего собственного бизнеса, или бизнеса ваших клиентов.

Вы уже затратили горы драгоценного времени и ресурсов на то, чтобы дать команде возможность как следует вникнуть в предметную область, сформировать концепцию системы и составить ее архитектурный каркас. Поначалу все шло как по маслу. Работа кипела, каждый день рождались новые схемы, планы, прототипы. Минимум раз в пару дней, собираясь в небольшие группы ребята находили неожиданные решения в ходе энергичных мозговых штурмов. Иной раз казалось, что охваченные творческим огнем, они работали как один человек.

Но однажды, что-то пошло не так… То, что прежде казалось могучей командой единомышленников, объединенных общей целью, внезапно превратилось просто в группу людей, работающих за зарплату. И подобно строителям Вавилонской башни, эти люди перестали понимать друг друга. Эксперты по бизнесу и специалисты предметной области, отгородившись стенами кабинетов и плотными графиками встреч, превратились в служителей культа, недоступного простым смертным. Разработчики же в ответ, недолго думая, перешли на свой привычный птичий язык, который и заложили в основание системы. Те же члены команды, кто призван проверять и гарантировать высокое качество, оказавшись на перепутье, стали походить на отдел ОТК, дающий заключения, когда кардинально исправить ситуацию уже нельзя.

Команда разделилась на несколько лагерей, в котором каждый создал свой язык. И, подобно пресловутым слепым из восточной притчи, каждый был уверен, что уж он-то знает, на что похож слон!

Ваша система тем временем продолжает деградировать.. Процессы, от которых зависит репутация и прибыльность вашей компании, оказались под угрозой серьезных сбоев. Примерно шесть месяцев назад уже начали прилетать первые ласточки. С того момента, число ошибок и негативных отзывов пользователей, несмотря на героические усилия команды, стабильно держится на том же уровне. Затыкая брешь в одном месте, вы порождаете ее в другом. Система вступила в состояние преждевременного старения, когда с каждым новым выпуском затраты на внедрение новой функциональности неуклонно растут, а качество падает. То, что раньше казалось простым и исчислялось часами и днями, теперь отнимает долгие недели, а то и месяцы кропотливой и не слишком творческой работы, без каких-либо гарантий окупаемости понесенных затрат.

Более того, если вашему бизнесу вдруг понадобится кардинально новая функциональность и вам потребуется срочно усилить команду новыми разработчиками, вы не сможете этого сделать. Ведь длительность введения в проект новичков составляет теперь минимум два месяца. При этом разработчикам придется изучить не только язык предметной области, но и «птичий язык», заложенный в систему старшими собратьями по цеху, часть из которых уже покинула компанию. Таким образом, вы не только тратите лишнее время на обучение, но и загоняете себя в ловушку зависимости от ключевых сотрудников. Ведь только они знают, как на самом деле работает эта штука…

Все понимают, что дальше так жить нельзя. Ведь если продолжить откладывать решение на завтра, рано или поздно наступит момент, когда невозможно будет сделать и шаг, чтобы не наступить на мину. Но как это сделать когда с каждым выпуском нехватка времени ощущается все больше и больше, а единство в команде – все меньше и меньше.

*Как вернуть в команду забытое единство и разрушить башни лагерей?*

*Как научить бизнес, команду и код системы говорить на одном языке?*

*Как снизить затраты на разработку и повысить качество?*

Предметно-ориентированное проектирование (Domain-Driven Design, DDD) – уникальный подход к внутри-командному взаимодействию и построению программных систем, находящийся на стыке программной инженерии, менеджмента и социальной психологии. Он помогает вашей команде, - всем и каждому, вне зависимости от его роли (эксперт по бизнесу, специалист предметной области, разработчик, инженер по тестированию) вдохнуть в проект свежие силы, вернув в него утраченные прозрачность и чувство цели. Отныне все участники будут учиться говорить друг с другом без переводчиков на одном языке, и одновременно строить на нем всю систему – языке вашего бизнеса.

Без тайного знания и черной магии... Только чистая предметная область. Основные сущности и бизнес-правила системы, важные сценарии и элементы интерфейса системы – все понятно и привычно как бизнес-экспертам, так и разработчикам системы.

Архитектура системы и ее исходный код идеально соответствуют созданной командой логике предметной области и ложатся на нее как перчатка. Они прозрачны, естественны и с минимальным инструктажем могут быть прочитаны даже теми, кто мало знаком с программированием – от девочки-дизайнера до директора фирмы.

Новые функции не просто механически добавляются по запросу заказчика, а осознанно прорабатываются командой и работают на лучшее понимание системы и бизнеса организации. При этом, за счет командой синергии, разработка и внедрение происходит значительно быстрее и с лучшим качеством.

Новые люди, приходящие в проект (вне зависимости от их специализации) теперь могут приступить к продуктивной работе не через недели и месяцы, а уже через несколько дней. Поскольку теперь язык системы точно соответствует языку бизнеса, теперь нужно изучить лишь последний. А так как, за счет упрощения, из системы ушли лишние взаимосвязи, вместе с ними ушли и лишние риски. Таким образом, объем необходимой для работы информации сократился в 3-5 раз. Это сделало команду не только сильнее, но снизило риски зависимости от ключевых сотрудников, которые теперь превратились просто в главных экспертов.