



hardsoft321

BIZOPSDEV КРАТКОЕ ВВЕДЕНИЕ



ПРОСТОТА ЦЕЛЕЙ КОРПОРАТИВНОГО ИТ

• и проверка на практике различных подходов

Цели корпоративного ИТ в целом остаются неизменными: улучшение условий и эффективности работы для клиентов и внутренних пользователей, улучшение коммуникаций между ними, снижение общих операционных потерь организации и общей стоимости владения ИТ, а также эффективное использование уже имеющегося ИТ-потенциала.

Начиная с середины 80х и по сегодняшний день, проявился только один подход, который давал раз за разом устойчивый положительный результат.

Мы обобщили его и оставили самую суть. Он получился очень простым.

Встречайте!

bizopsdev³²¹



СУТЬ ПОДХОДА

• предельно кратко



Давайте объединим заинтересованных ключевых специалистов из бизнеса* **biz-** с ИТ-специалистами, которые сопровождают **-ops-** и развивают **-dev** используемые данным бизнесом ИТ, в одну небольшую команду** на границе с Общим ИТ***. И давайте не допустим в дальнейшем разрыва между бизнесом **biz-** и ИТ, а также внутри самого ИТ между сопровождением и развитием **-opsdev**.

Эта команда будет отвечать за ВСЕ*** вопросы, связанные с использованием и развитием ИТ для данного бизнеса. Все вопросы – это означает все от проблем с техникой на рабочем месте до функциональности специализированного программного обеспечения и собственного личного состава команды.

Это и будет команда **bizopsdev**.

Уже в среднесрочной перспективе, через полгода-год, от такой команды можно ожидать:

- Приоритизацию и выполнение задач, которые реально сразу облегчают работу и взаимодействие сотрудников и клиентов.
- Исключение неактуальных задач из списка сформированного до создания этой команды.
- Большинство задач (80% и более) будут решаться самой командой без обращения за внешней помощью.
- Стабилизацию само-сформированного состава команды.

* или бизнес-подразделения в зависимости от размера бизнеса/организации.

** порядка 7+-2 человека, на самом старте можно и меньше.

*** если Общее ИТ уже сформировалось. См. также стр. 7 Greenfield vs Brownfield.

**** режим “одного окна”.

РАЗМЕР ИМЕЕТ ЗНАЧЕНИЕ

• Как будем масштабироваться?

1. Что делать, если Общее ИТ состоит из сотен, а то и тысяч сотрудников?
2. Что делать, если компания слишком мала, чтобы включить в состав bizopsdev-команды хотя бы двух-трех программистов?
3. Что делать, если бизнес начнет активно расти, и пропорционально будет расти нагрузка на bizopsdev-команду(ы)?

1. На старте под каждый значимый центр доходов и затрат создается своя bizopsdev-команда, которая работает в режиме «одного окна». Дальнейшая трансформация bizopsdev-команд и других команд* Общего ИТ будет происходить по ходу работы.
2. Необходимо воспользоваться услугами подходящей команды у ИТ-подрядчика в режиме Team-as-a-Service, начиная с подписки на одного универсального виртуального специалиста.
3. Каждая bizopsdev-команда будет увеличиваться по мере роста нагрузки в два раза по размеру, а затем разделяться на две самостоятельные команды. Критерии разделения нагрузки и функциональности определяются на месте в соответствии с контекстом роста нагрузки на ИТ. Удвоение числа bizopsdev-команд возможно каждые полгода.

* * * * *

* * * * *

* * * * *

* примеры других команд можно посмотреть в книге “Team Topologies”.

* * * * *

* * * * *

* * * * *

ЧТО МЕШАЕТ

распространению bizopsdev практики ?

Вот пара основных причин:

1. Уникальная особенность деятельности по разработке и сопровождению ПО и ее неочевидность*. Ближайшим аналогом для которой является научно-исследовательская лаборатория, а не потоковое производство или строительство зданий**. Психологический управленческий барьер для традиционного менеджмента при соприкосновении с высокой степенью автономности и самоуправляемости bizopsdev-команд.
2. Самым важным результатом работы bizopsdev-команды является формирование коллективного Когнитивного Облака*** знаний, навыков и личностных отношений как внутри команды, так и с ее окружением. Такое Когнитивное Облако обеспечивает принципиально другой уровень эффективности использования и контроля за Кодовой Базой и ИТ-инструментами. По степени значимости на первом месте идет Когнитивное Облако, затем Кодовая База, и лишь на последнем месте инструменты (используемый ИТ-стек). В следствие неочевидности (см. п.п.1), исторически так сложилось, что менеджмент оценивает значимость в противоположенном порядке****. Во многих случаях уникальная Кодовая База и уникальная ИТ-команда воспринимаются как источник проблем, рисков и дороговизны. Когнитивное Облако вообще отсутствует как понятие на уровне топ-менеджмента. Все это в корне неверно и контрпродуктивно.

* она неочевидна и большинству ИТ-специалистов. Cynefin Framework от Dave Snowden и принцип Team First из “Team Topologies”.

** по факту отрасль строительства зданий и конвейерное производство явились худшими аналогиями и нашли свое отражение в ИТ-терминологии.

*** так же формируется внутренний Универсальный Язык команды. “Team Topologies”

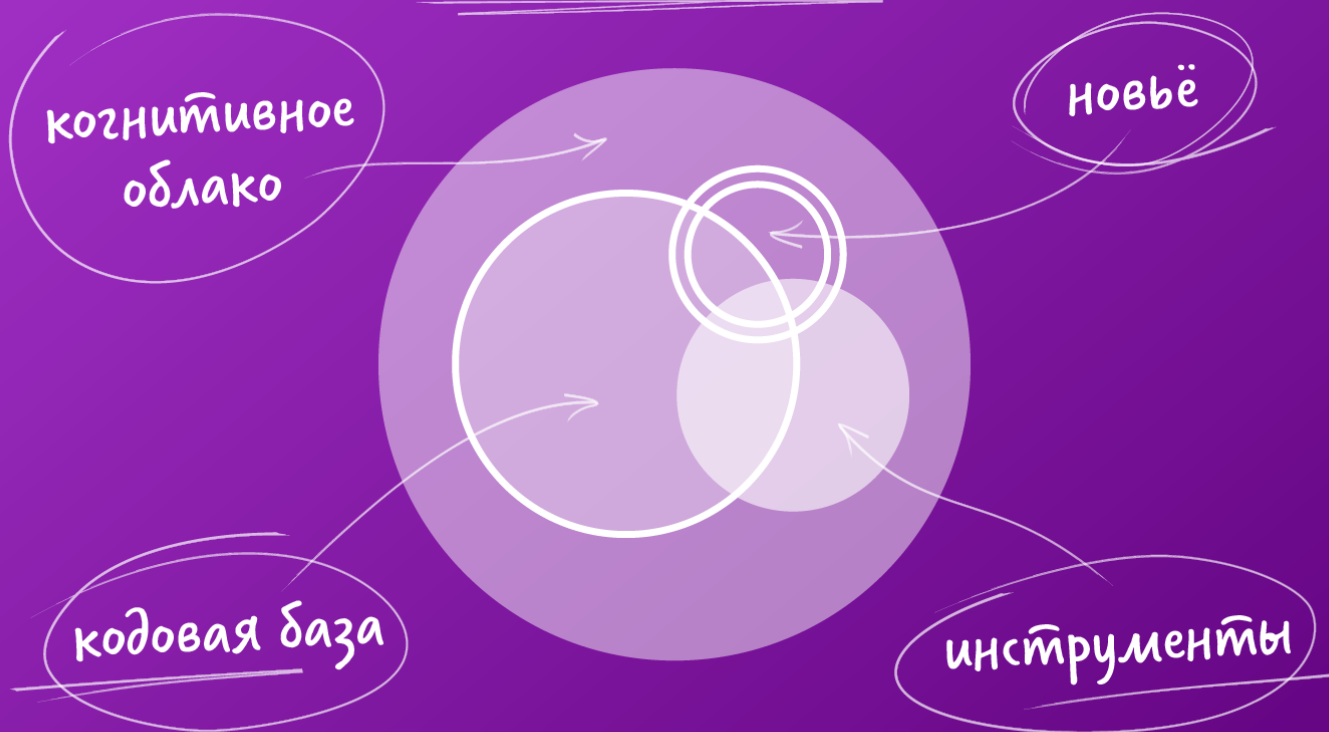
**** этим объясняется фиксация на ИТ-инструментах, которые часто называются платформами, готовыми системами „из коробки“ и промышленными решениями. На неудачный выбор ИТ-инструментария легче всего списываются ИТ-неудачи и трудности. Идет постоянное привнесение “новья” в существующий ИТ-ландшафт. И, как закономерный итог, такие ИТ-ландшафты становятся похожи на нагромождение устаревшего кода.

Посмотрите нашу поясняющую визуализацию на следующей странице!





bizopsdev³²¹



Размер каждого круга означает значимость и критичность для организации*.

На данном рисунке изображена идеальная ситуация, когда Когнитивное Облако, т.е. компетенции команды, полностью покрывает и Кодовую Базу** и Инструменты.

Более подробно данная визуализация будет разобрана в других материалах.



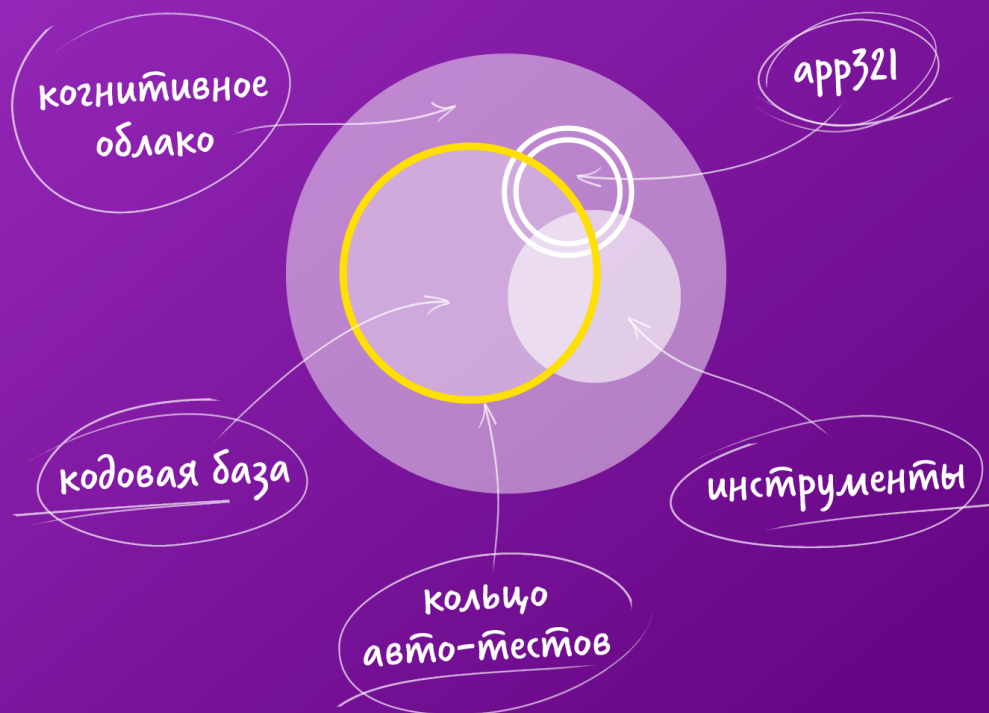
* Для простоты понимания и применения присвоим им соответствующие веса: 3 — Когнитивное Облако, 2 — Кодовая База, 1 — Инструменты. И получим, что наличие Когнитивного Облака, что по сути равно наличию bizopsdev-команды, контролирующей кодовую базу, в 5 раз важнее Инструментов (технологического стека), на котором реализовано решение.

** Кодовая База включает в себя: коды программ и авто-тестов, пользовательские данные, документацию.

ЧИСТЫЙ ЛИСТ или БОЛЬШАЯ ГЛИНА №4*

GREENFIELD
VS
BROWNFIELD

bizopsdev³²¹



Практически для всех средних и крупных организаций ситуация с «чистого листа»** осталась в прошлом. Когнитивные облака команд, как правило, не покрывают кодовую базу.

В такой ситуации восстановление данного покрытия и контроля над кодовой базой может быть обеспечено через кольцо*** кросс-функциональных пользовательских авто-тестов.

После этого, при необходимости миграции кодовой базы на новый технологический стек, мы рекомендуем начать с минимальной кодовой базы (например, с app321 или аналогичного СПО-решения) и асинхронной интеграции (Кafka) его на время перехода в единое целое с существующей кодовой базой****.

* Просто погуглите это.

** см. информацию на тему Greenfield vs Brownfield.

*** отсылка к фильму «Властелин Колец» и главному кольцу, «чтобы управлять ими всеми».

**** тем самым мы обойдем тяжелейшее ограничение Закона Галла для сложных ИТ-систем.

ПРОСТОТА ПРИМЕНЕНИЯ

• и ключевой фактор снижения риска потери команды

Важным преимуществом подхода bizopsdev является его простота применения.

Однако после более чем 30 лет его использования мы обнаружили один критически важный фактор. Риски потери такой команды драматически снижаются, если она разделена на две части: opsdev между бизнес-компанией и полностью независимым ИТ-подрядчиком.

Это отражено в логотипе bizopsdev^321.

bizopsdev **321**
SELF-ORG TEAMS ONLY SINCE 1990

Желаем успехов и да пребудет с вами сила bizopsdev-команд!