**Основные законы создания команд разработчиков**

Закон Брукса

Ни одно обсуждение команд разработчиков не проходит без упоминания данного принципа:

«Добавляя людских ресурсов, мы задерживаем окончание программного проекта» (Брукс, 1975).

Бесчисленные команды разработчиков подтвердили постулат. Законы Брукса и Конвея составляют базу.

Присоединяя нового человека, команда тратит усилия на введение в курс дела, на объяснение используемых трюков и устройства. Участники расходуют время на информирование и синхронизацию с новобранцем, на обучение труду в команде и передачу знаний. Работа замедляется.

Свежий персонал нуждается в помощи не только в первую неделю. Часть авторов (например, Коплиен и Харрисон) предполагают годичный интервал до получения от новичка продуктивности. Не стоит придавать утверждению излишнего значения из-за влияния разных факторов. Разумно предположить, что часть сотрудников обойдется без помощи в первые три месяца.

Команда может замедлить работу задолго до появления нового человека. Сотрудник не возникает просто так. Иногда менеджеры по персоналу действуют в частных интересах, запрашивая больше ресурсов. Должностные инструкции нужно написать, проверить, издать, перенаправить в отдел кадров, послать рекрутским агентствам. Потребуется время на утверждение процессов.

Резюме следует обдумать и написать либо отказ, либо вызов на собеседование. В случае одобрения кандидата составляется контракт и отправляется предложение о приеме в команду.

Процедуры проходят до получения человеком права написать строчку кода. Даже если отдел кадров руководит большинством процессов, лидеры команды и простые члены обязательно будут отвлекаться. Время на реальную работу и развитие сократится.

Закон Брукса не означает полный отказ от расширения команд. Но рост объединения редко происходит быстро. Если команда решила расширяться, будет использоваться часть производительности для наращивания мощности.

Ричард Шеридан сделал смелое заявление:

Я рад сообщить, что закон Брукса может быть нарушен. Вся наша деятельность сфокусирована на том, чтобы сломать утверждение. Разделение людей на пары, перемещение между парами, автоматизация тестирования, управление кодом, наём, основанный не на героях, постоянные переговоры, открытая рабочая среда и видимые артефакты — всё, чтобы опровергнуть утверждение Брукса (Шеридан, 2013).

В книге автор преподносит среду разработки ПО отличную от реальной для большинства разработчиков и менеджеров. Корпорация Menlo не скупится в вопросах построения, участия и укрепления собственной культуры и сообщества. Пока компании не опробуют подход на практике, не возникнет достаточное количество примеров для изучения, трудно сказать, стоит ли копировать образцы Шеридана.

Методы, описанные автором, заслуживают доверия. Отлично, если закон Брукса будет нарушен.

Закон Конвея

Организации, проектирующие системы, … производят их, копируя структуры коммуникации, сложившиеся в этих организациях,

 *Конвей, 1968*

Люди и их организация играют значительную роль в определении архитектуры ПО, принимаемой разработчиками.

Представим создание правительством новой системы социального обеспечения. Потребуются базы данных, интерфейсы, бизнес-логика. Понадобятся разработчики и тестировщики, дорогостоящий менеджмент, установка требований и прочее.

В итоге результат обретает видимые очертания: определены роли пользователей, workflow, формы данных, отчеты. В этот момент, согласитесь, создание меньшей или принципиально отличающейся системы уже невозможно.

Взглянем на систему после 10-ти лет функционирования. Наблюдается обратный закон Конвея:

Предприятия, использующие программные системы… ограничены структурами коммуникации, которые копируют эту систему.

Закон Конвея сообщает о возможном копировании проблем предприятия в интерфейсе: в слоях, или в APL, или в модулях, или где-нибудь ещё.

Закон Конвея необходимо учитывать при организации команд и софтверных систем. Попытки сломать принцип сознательно или по неведению создадут силы противодействия проекту. Выглядит, как желание расколоть древесину поперек волокон. Разумнее уважать и использовать закон Конвея.

Число Данбара. Природные ориентиры

«Экстраполяция на людей отношений среди обезьян даёт представления о размерах социальных групп. Около 150 особей — предел социальных отношений человека. Количество удостоилось названия «Числа Данбара»,

 Данбар, 2010

Частый вопрос от групп разработчиков: «Насколько большой должна быть команда?» Работа антрополога Робина Данбара дает интересные идеи при попытке ответить на вопрос.

Автор предоставляет убедительные доводы, что 150 является верхним пределом для организациилюдей. Данбар заметил указанное число в воинских формированиях со времён Древнего Рима, в неолитических поселениях, в общинах амишей и в современных научно-исследовательских группах.

Сообщества свыше 150 членов менее сплочённые, требующие повышенного контроля поведения и иерархии. В исследованиях и анализе подчёркиваются основные моменты в формировании групп. Параметр грамотнее назвать «Числами Данбара». Характеристики применимы к различным группам, входящим в состав крупных формирований. Меньшие команды крепче и по предположению опираются на числа с множителем 3.

Следуя тезису, у большинства людей круг близких друзей от 3 до 5 человек. Следующий уровень приятелей от 10 человек до 13–15 (при определенных усилиях). Очередная группа содержит от 30 до 50 — типичный боевой взвод. Популяция в 150 представляет минимальный независимый блок в военной компании и точку создания на предприятиях отдельных группировок.

Данбар предполагает существование формирования в 500 и 1500. Автор поддерживает Платона в определении идеального размера для демократии в 5300 единиц. Можно провести интересные параллели между размерами военных блоков.

|  |  |
| --- | --- |
| Oрганизация | Размер |
| Пожарная дружина | 4 или меньше человек |
| Секция/Отряд | 8–12 участников (несколько пожарных дружин) |
| Взвод | 15–30 служащих (2 отряда) |
| Рота | 80–250 солдат (несколько взводов) |
| Батальон | 300–800 бойцов |

Список не является конечным. Не стоит забывать о различиях между странами. Даже об особенностях флангов одной военной организации. Но в широком смысле размеры блоков следуют выводам Данбара.

Магическая семерка Миллера (кошелёк Миллера)

Принято считать мудрым выбором размер команды из 7 человек (± 2). Практического смысла в утверждении нет. Доказательства тезиса об оптимальном количестве членов команды в 5–9 человек отсутствуют.

Сторонники размера апеллируют к знаменитой статье Миллера 1956 года «Магическое число семь плюс минус два: некоторые ограничения нашей способности обработки информации». На практике большинство ссылающихся труд не читали.

В статье Миллер утверждает, что 7 является важнейшим числом для описания мощности обрабатывающих возможностей человеческого мозга. Выбранная цифра определяет максимальное количество «кусков» информации для одновременной обработки мыслительным центром. Автор приходит к выводу о неоднозначности трактовки частого повторения числа 7:

«В настоящий момент я призываю воздержаться от однозначных суждений. Возможно, в глубине этих семерок скрывается что-то важное, что необходимо открыть. Но я подозреваю, что все может оказаться лишь пагубным Пифагоровым совпадением».

Значимость магии числа 7 под вопросом.

В заключении автор говорит: «Я чувствую, что мой рассказ здесь должен остановиться, т. к. он уже стал достаточно интересным». С публикации статьи Миллера прошло более 50 лет. По общему мнению, диапазон команды 5–9 человек удовлетворителен для управления изменениями. Снизу интервала оправдана необходимость проведения тестов и выполнение требований к команде специалистов. На верхней границе диапазона сложно осуществлять полный контроль системы. Поэтому число сотрудников от 5 до 9 человек имеет смысл.

Описанное количество не отменяет существование команд большего размера в зависимости от обстоятельств.

Размер команды в методологии Scrum

Как статья Миллера об объемах обработки информации человеческим мозгом может применяться к определению размера команды разработчиков ПО? Обратимся к методологии Scrum. В учебнике говорится: «Команда в Scrum должна быть семь плюс или минус два человека» (Димер и др., 2008). Одновременно в руководстве по Scrum за 2011 утверждается: «Команды из более девяти членов вызывают слишком много проблем в координации. Большие команды разработчиков заметно усложняют весь процесс» (Сазерленд и Швабер, 2013).

Учебное пособие гласит:

Роли владельца продукта и руководителя команды Scrum не включены в подсчет, кроме случаев, когда они включаются в работу для сокращения отставания в очередном спринте,

— Сазерленд и Швабер, 2013

Параллельно введение руководителя команды Scrum предполагает, что владелец продукта находится вне команды.

Другие источники дают различные рекомендации по определению членов группы и «просто участвующих».

Команды в диапазоне 4–8 человек видны повсеместно. Статьей Миллера рационально обосновать выбор размеров групп из 7 ± 2. Опыт подтверждает оптимальный предел численности. Количество может быть больше заявленного в источниках по Scrum.

Закон Паркинсона и Закон Хофштадтера

Работа заполняет все время, выделенное для ее выполнения,

*— закон Паркинсона*

Всегда потребуется больше времени, чем вы ожидаете, даже если вы знаете закон Хофштадтера,

*— закон Хофштадтера, 1980*

Попробуем ответить на вопрос: «Помните ли вы обучение в ВУЗе? В какой момент реализовывались задания?» Большинство читателей выполняли работу за несколько дней до дедлайна. Часть занималась проектом в ночь перед сдачей.

И подавляющее большинство укладывалось в срок.

Ученые подтверждают: люди плохо справляются с оценкой периода, требующегося на выполнение работы, но преуспевают с решением задачи к четко оговоренной дате (например, Buehler, Гриффин и Peetz, 2010).

При избытке времени на проект происходит чрезмерное расширение объема работ исполнителем.

На разработку программного обеспечения влияют законы Паркинсона и Хофштадтера. Выбирая дату дедлайна, неизбежно возникнет недооценка требуемого времени. В итоге сроки и объем работ увеличатся. Проведенное исследование (Бюлер, Гриффин и Питз, 2010) свидетельствует: оптимизм относительно даты выполнения задачи дает возможность осуществить работу раньше, чем при пессимистичной оценке сроков. Но общее время выполнения проекта оптимистом окажется дольше, чем у пессимиста. Срок дедлайна может быть важнее предполагаемого времени завершения проекта (Арили и Wertenbroch, 2002).

Закон Голла. Парнас и Александер

Сложная рабочая система неизменно получается из простой рабочей системы. Сложная система, разработанная с нуля, никогда не работает. И никакие улучшения не заставят ее работать. Начинать следует с простой рабочей системы,

*— закон Голла, 1986*

Закон перекликается со словами Дэвида Парнаса:

«Как правило, системы ПО не работают хорошо, пока они не были использованы, и не раз, в «боевых» условиях».

Авторы подчеркивают различные аспекты одного явления, называемого Кристофером Александером «органическим ростом».

При создании ПО, применяя технику под названием «ходячий скелет», рекомендуется сделать простой, базовый рабочий кусок кода. «Скелет», хоть и с определенным риском, но проталкивающий систему вперед. Создав базу, команда добавляет «плоть» — слой функционала.

Принцип может применяться к группам, к ПО:

«Эффективная сложная команда неизменно возникает из простого продуктивного аналога. Сложная команда, собранная с нуля, результативно функционировать не может. И никакие изменения не заставят ее работать. Дело стоит начинать с уже сработавшейся командой».

Можно провести параллель между сложной и крупной командой. Закон намекает на способ создания больших групп, на гибкую методологию масштабирования объединений.

Очевидна связь с законом Конвея: построение «ходячего скелета» базируется на костяке команды. Для создания большого и сложного формирования используется равноценный скелет-костяк.

При построении изначально масштабной структуры Закон Конвея предрекает большую и сложную архитектуру. Закон Голла говорит о появлении нерабочей системы и для выдерживания сроков советует команде начать с меньшего.

Законы Келли

Следует добавить пару полезных постулатов от автора статьи.

Масштаб ПО всегда будет увеличиваться пропорционально имеющимся ресурсам,

*— первый закон Келли*

Внутри каждого большого проекта в области разработки есть маленький побочный проект вне основной задачи

*— второй закон Келли*

Излишне масштабная команда для оправдания собственного размера выдаст больший объем трудозатрат, громоздкие решения и мудреную архитектуру. Легче добавить, чем удалить против воли, участника группы.

Сохраняя размер команды небольшим хотя бы на начальном этапе, есть вероятность найти простое и лаконичное решение. Старт с большой командой будет гарантировать громоздкую реализацию.

Вывод

Числа Данбара определяют лимит размера эффективной команды. Опираясь на закон Конвея, можно говорить о потенциальном пределе системы. Единственный способ обойти аксиому — разделить большую систему на компактные группы. Команды не рождаются полностью сформированными и эффективными. Закон Конвея работает в совокупности с тезисом Голла. Группы должны расти постепенно. Постулат Брукса подразумевает, что команды не могут расшириться слишком быстро. Закон Паркинсона означает занятость излишне крупных команд поддержанием собственного существования.

Второй постулат Келли подсказывает решение: избегайте большого, сосредоточьтесь на малом.

Лучше не противостоять законам, а найти подход и работать с положениями. Попытки найти способ обращения с законами могут быть коммерчески выгодными, если использовать их хотя бы для обоснования принятых решений перед вышестоящими начальниками.