

Capacity — это что?

Для начала давайте разберемся, что включает в себя понятие «капасити», для чего оно нужно и с чем его, простите, едят.

Капасити (англ. “capacity”) — это показатель максимальной ёмкости чего-либо. Например, в IT капасити можно применить в контексте ресурсов (штат, техника и так далее) или в контексте пользователей, конечного/будущего потребителя продукта.

Как это применить на проекте?

Методика расчета

Чтобы понять, как рассчитать капасити команды, предлагаю рассмотреть этот процесс на примере команды из 18 человек, в которой каждый сотрудник работает ежедневно по 8 часов.

Для подсчета капасити мы ориентируемся не на 8, а на 6 часов, закладывая 2 часа на допустимые внутренние риски проекта.

Дальше мы оцениваем в процентах, сколько каждый человек работает над фичей. Здесь нам важно учитывать конкретно время работы над фичей, исключая созвоны и обсуждения по задаче.

Опираясь на опыт SimbirSoft, мы можем сказать, что у сотрудника уровня Middle+/Senior отдача проекту будет на уровне 90-100%. Все остальные специалисты уступают по процентовке.

Далее для удобства в подсчете переменную в 100% принимаем за 1. Соответственно, если отдача сотрудника меньше 1, то это будет 0,9, 0,8 и так далее.

На созвоны мы закладываем 30% только у лидов, если они есть. Так происходит потому, что все вводные проходят через лидов, и конечному специалисту в работу приходит уже декомпозированная задача с минимумом неопределенностей. Производительность лидов на проекте тем самым снижается до 0,7.

Дополним, что 100% производительность на проекте может быть только у специалистов с опытом работы от 3-5-ти лет. Но это ориентир, который мы обозначили на основе нашего опыта. Поэтому на других проектах возможны исключения.

Также важно не забыть обязательные условия — это ограничения в работе, в которые входят отпускные и больничные дни.

Расчет капацити команды

Теперь предлагаю все расчеты применить на примере нашей команды.

2 аналитика: Lead (0,7) и Middle+ (1)

Дизайнер: Middle+ (1)

Frontend IOS: Lead (0,7) и 2 Middle+ (2), Android: Lead (0,7) и 2 Middle+ (2)

Backend: Lead (0,7) и 2 Senior (2)

QA: Lead (0,7) и 3 Middle+ (3)

AQA: 2 Middle+ (2)

Удобно, что используя этот метод, мы можем оценить капацити команды на любой период от одного дня. Кроме того, оценивая загрузку с помощью капацити в человеко-днях, можно оценить любую команду на любой срок.

В месяце 21 рабочий день, а в спринте — 10 рабочих дней. Предлагаю рассчитать капацити команды для спринта, учитывая, что никаких дополнительных ограничений нет:

$$\text{Аналитика} : (\textit{Lead} + \textit{Middle+}) * \text{количество рабочих дней} = (0.7 + 1) * 10 = 17$$

По аналогии рассчитаем другие направления:

Design: 10

IOS: 27

Android: 27

Backend: 27

QA: 37

AQA: 20

Итого kapasitas команды на спринт равен 165 человеко-дням.

Фича приходит на оценку тоже в человеко-днях, с учетом ограничений kapasitas отдельных направлений и команды в целом.

Таким образом, мы можем понять, в каком направлении и насколько у нас недостаточная загрузка по команде, и рассмотреть варианты перекинуть специалиста в другую команду или добавить в бэклог спринта еще одну фичу.

Ограничения в расчете kapasitas

В этом методе работы с kapasitas команды есть несколько ограничений, которые важно зафиксировать, и отметить, к чему они могут привести:

1. Мы измеряем все человеко-днями и в оценке фич, и в подсчете kapasitas. Если мы измеряем объемы работы разными переменными, то свести их воедино будет проблематично — каждая будет жить своей жизнью.
2. Благодаря такой методике мы имеем прозрачную схему работы каждого сотрудника на ближайший период, на который мы рассчитываем kapasitas, будь то неделя, две или месяц.
3. Kapasiti — величина динамическая. Мы должны работать с верными и точными данными на проекте. Если все-таки кто-то уходит на больничный или в отпуск, то kapasitas пересчитывается. Благодаря этому мы формируем правильные ожидания у заказчика. Впоследствии это приведет только к улучшению понимания в цепочке «Заказчик — менеджер проекта — команда».
4. В подсчете kapasitas уже учтены созвоны по фичам, активности лидов, но не учтены работы багофикс, техдолги и прочее. Исходя из особенностей проекта, его срока жизни, количество необходимых фиксов и задач техдолга может варьироваться от нескольких десятков до нескольких сотен. Индивидуальность в подходе их исправления приветствуется, но эту особенность нужно закладывать как отдельный пункт в бэклоге задач.

Вывод

В этой статье я постаралась максимально подробно и понятно расписать, как можно рассчитать kapasitas конкретной команды на нужный промежуток времени и поделилась ограничениями, которые могут возникнуть.

Возвращаясь к цели внедрения kapasitas конкретно на нашем проекте, то на сегодняшний день мы имеем следующие результаты:

1. Производительность команды стабильна и составляет свыше 90% по выполнению запланированного плана работ на спринт.
2. С учетом ограничений мы формируем полноценную загрузку команды по всем направлениям, к минимуму сводим простой сотрудников в спринте.
3. Внедрение капасити добавило дополнительную ценность как для команды, так и для заказчика в виде прозрачности загрузки, а также возможности приоритизации задач в бэклоге на следующие спринты.

В нашем случае весь процесс внедрения занял 4 месяца. На это повлияло в большей степени то, что спринт на проекте длится в течение одного месяца, в команде около 20 человек — по 3-4 человека от каждого направления. Для небольших команд, например по 10 специалистов, для внедрения будет достаточно и одного-двух спринтов, продолжительностью 2 недели. Это позволит понять, насколько такой подход работы удовлетворяет потребности и что стоит изменить в нем. Поделитесь в комментариях, используете ли вы капасити в своей команде, и как это повлияло на производительность.