

# Раздел 10. Методология Scaled Agile Framework (SAF)

## Как мы масштабировали agile в тестировании: обзор Scaled Agile Framework

*Предлагаем поговорить про гибкие методологии. Но речь пойдет не про Scrum, с которым многие знакомы не понаслышке, а про SAFe – методологию, которая позволяет делать гибким процесс разработки ПО в очень больших командах.*

Начнем с предыстории. У нас есть заказчик, качество продуктов которого мы обеспечиваем уже более четырех лет. Работа над его решениями строилась по Scrum'у до тех пор, пока однажды он не поставил нас перед фактом: с завтрашнего дня переходим на SAFe. Данный фреймворк можно отнести к редко используемым, и мы решили не упускать возможности детально его изучить.

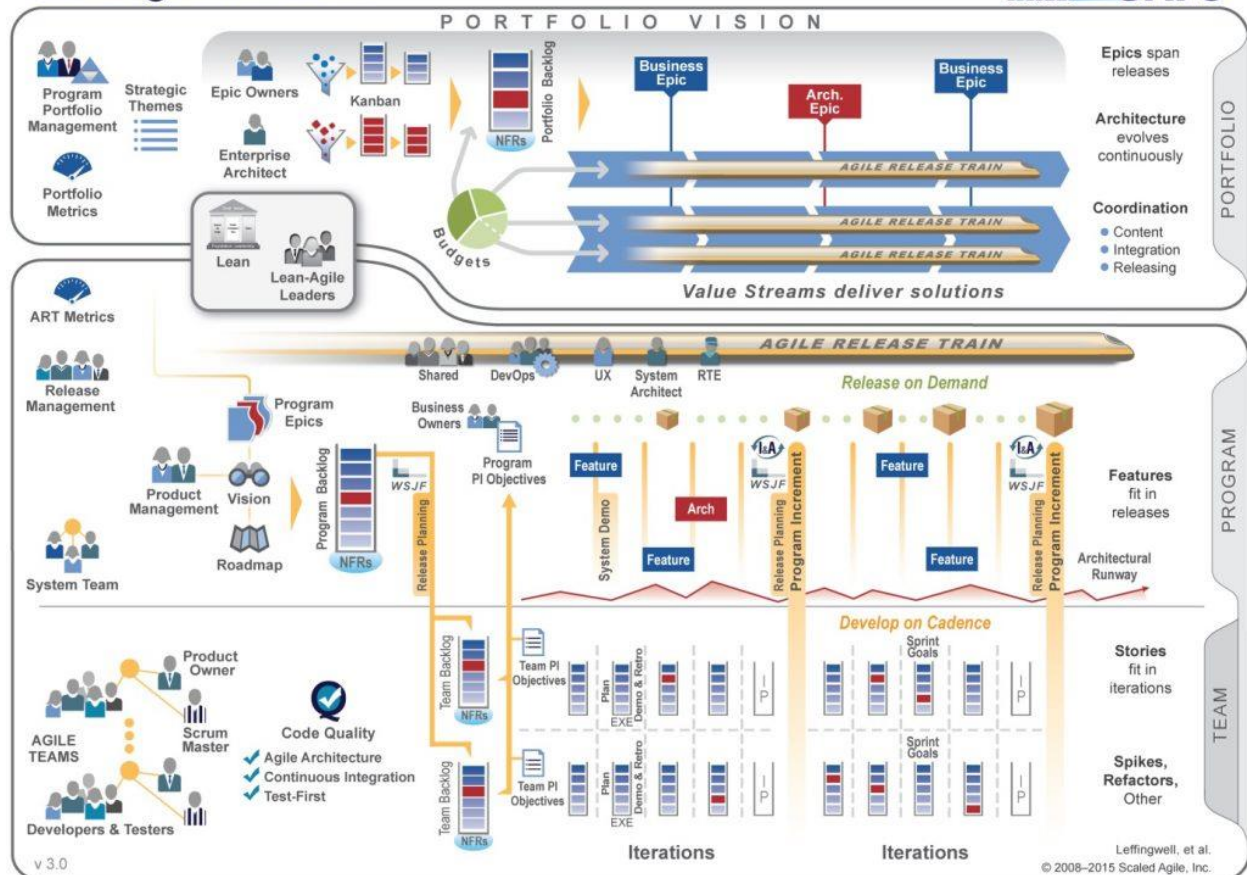
Внедрение SAFe происходило методом проб и ошибок, но в итоге процесс был настроен, оптимизирован и уже более полутора лет успешно работает.

Мы решили поделиться интересным опытом и надеемся, что наши советы будут полезны и предупредят сложности, которые могут возникнуть у вашей команды.

В этой и еще четырех статьях мы дадим базовое представление о SAFe, расскажем о том, какие уровни он включает. Кроме этого, поделимся своим опытом внедрения фреймворка и перечислим главные отличия от обычного Scrum'а. Итак, начнем.

### Что такое SAFe?

На запрос «Что такое SAFe?» в Google мы получили вот такую диаграмму:



**Scaled Agile Framework** – это набор правил и предписаний для внедрения agile-принципов в больших организациях. Одной из причин популярности данного фреймворка сегодня – это неудачный опыт использования agile-принципов на крупных предприятиях, которые, с одной стороны, стремятся соответствовать принципам agile, а, с другой стороны, хотят добиться предсказуемости сроков выпуска продукта. Всем им необходим проверенный подход, который можно было бы «развернуть» среди многочисленных команд разработки.

## Когда стоит внедрять SAFe?

Наличие следующих предпосылок говорит о том, что на предприятии можно подумать о внедрении SAFe:

1. Множество разрабатываемых продуктов (так называемый портфель проектов).
2. Множество проектных ролей.
3. Сложная процедура внедрения какого-нибудь предложения или изменения (требуются десятки согласований).
4. Большое количество команд разработки и желание работать по Scrum.

Допустим, нам нужно увязать между собой разработку и релиз более чем 15 программных решений, каждое из которых имеет более десятка stakeholder'ов. Решения разрабатываются 10 командами из 5 стран. Такого уровня задачу решить с помощью традиционного Scrum'а невозможно.

### **Какое решение предлагает SAFe?**

#### **SAFe разбивает любое предприятие на три уровня:**

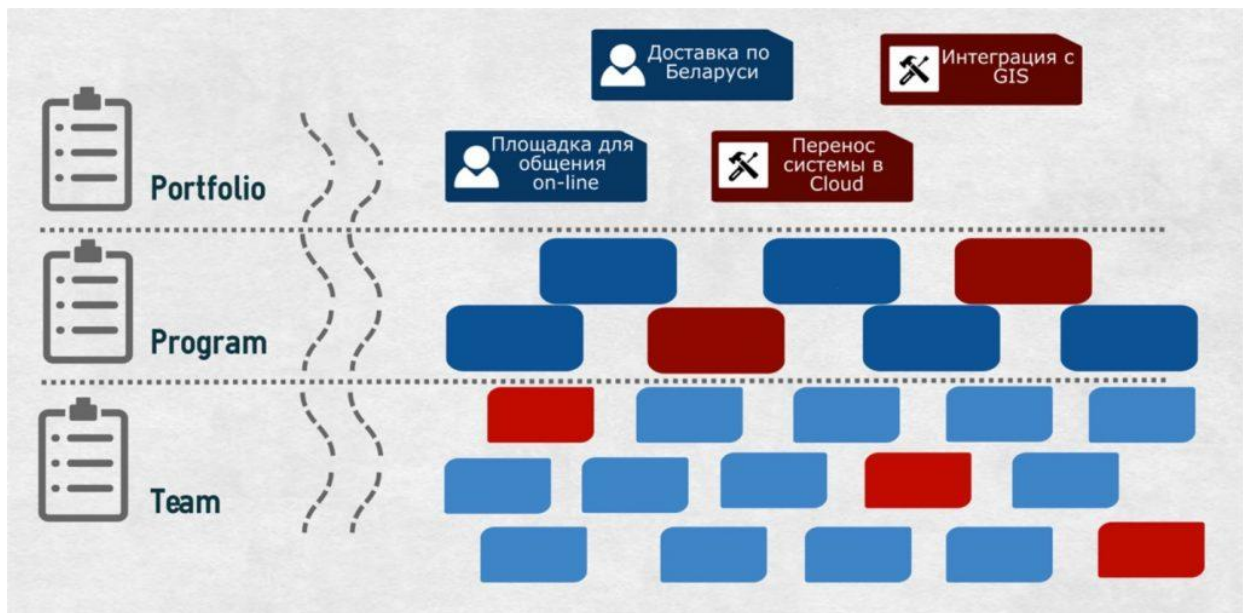
- уровень портфолио;
- уровень программ;
- уровень команд.

Давайте разберемся на примере простого вымышленного проекта, что собой представляет каждый из уровней.

Допустим, нам нужно доработать веб-приложение для книжного магазина. Основные функции текущей версии просты: на сайте через поиск можно найти книгу и оставить предзаказ. Когда заказ готов, менеджер связывается с покупателем, и тот самостоятельно забирает его со склада. Очевидно, что функциональность описанной версии веб-сайта весьма ограничена. Чтобы добавить конкурентных преимуществ, нужно придумать дополнительную функциональность на сайт.

Работа начинается на уровне портфолио. Здесь центральным является понятие «бизнес-эпика», которое отражает наиболее приоритетные направления развития продукта в ближайшее время. Эпики бывают функциональные и архитектурные.

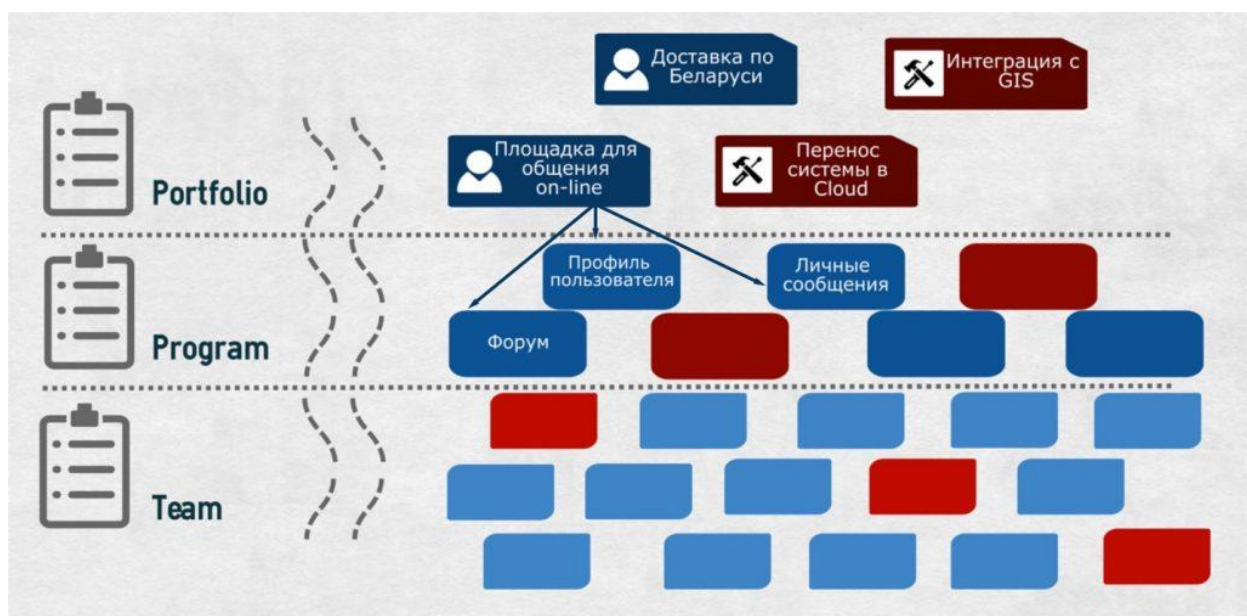
Для нашего веб-сайта примерами функциональных эпиков будут следующие: «Организовать доставку по стране с возможностью отслеживания обработки заказа в личном кабинете» и «Создать онлайн-площадку для общения любителей книг». Примерами архитектурных эпиков могут быть «Интеграция с геоинформационными системами» и «Перенос системы в облако».



### SAFe: пока все просто, спускаемся ниже

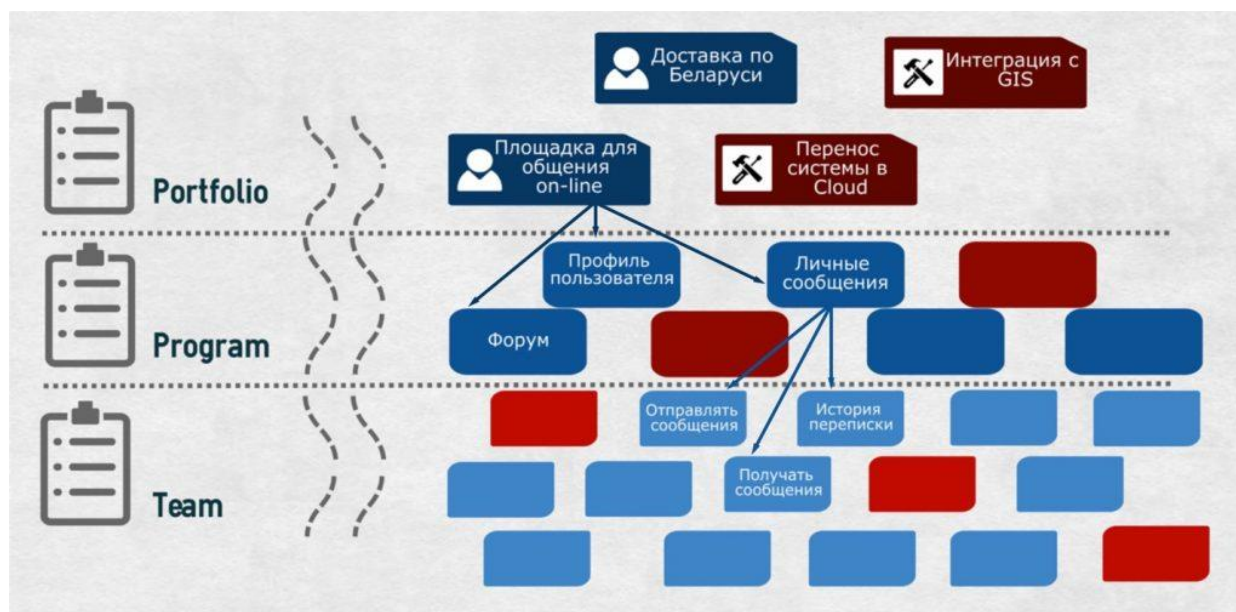
На уровне программ эпика разбиваются на фичи, т.е. куски функциональности, реализующие данный эпик. Фича – это ограниченный, чаще всего крупный кусок функциональности, который приносит определенную пользу бизнесу или позволяет конечному пользователю решать какую-либо задачу.

Давайте возьмем один из эпиков и детализируем его до уровня команд. Например, функциональный эпик про создание площадки для общения онлайн. На уровне программ он будет представлен фичами «Управление профилем пользователя», «Форум» и «Личные сообщения».



На уровне команд требуется еще большая декомпозиция задач. Каждая фича должна быть разбита на несколько понятных и четко сформулированных пользовательских историй (user stories), которые можно оценить и реализовать в течение спринта.

Например, фичу «Личные сообщения» можно декомпозировать в следующие пользовательские истории: «Возможность отправлять сообщения», «Возможность получать сообщения», «Сохранение истории переписки». Такая же декомпозиция потребуется и для архитектурных эпиков.



Таким образом, по мере спуска с уровня портфолио к уровню команд, задачи уменьшаются, а их границы уточняются. Соответственно, оценки также становятся точнее.