

Санкт-Петербургский национальный исследовательский
университет

информационных технологий, механики и оптики

Факультет программной инженерии и компьютерной техники

Лабораторная работа № 3 по дисциплине

«Экономика программных систем»

Методология **scrum** / VCS **git**

Выполнили:

Студенты группы Р43131

Ибадуллаев Алибаба

Осипов Василий

Преподаватель: Машина Е. А.

Санкт-Петербург

2023

Задание

Предложить план действий в ситуации, когда прошло 3/4 срока, запланированного на реализацию проекта, а фактически выполнена только половина задач:

1. Определить, какие функции на данный момент еще не завершены и оценить, реализацию каких из них можно отложить для того, чтобы не сдвигать срок выпуска устраивающего заказчика работоспособного продукта с максимально сохраненной функциональностью.
2. Оценить возможность увеличения команды разработчиков для соблюдения сроков проекта, либо попытаться оптимизировать план работ

Вариант

Сайт <http://mangalib.me>, git, scrum

Выполнение

Оценка сроков выполнения

Вспомним все поставленные в начале проекта задачи:

№	Функционал	Оценка , мин. /чел. час	Оценк, вероятн . /чел. час	Оценка, макс. /чел. час
1	Раздел каталога	30	40	50
2	Раздел просмотра информации о тайтле	15	20	30
3	Раздел комментариев	20	30	40
4	Раздел оценок	10	20	30
5	Раздел просмотра обсуждений	15	20	30
6	Раздел редактирования манги (описание/главы)	10	20	30
7	Раздел списков (читаю/ в планах/ избранное и т.д.)	10	15	20
8	Режим чтения	40	50	60
9	Раздел регистрации авторизации	10	15	20
10	Раздел личного кабинета	20	30	40
11	Раздел главной страницы	15	20	25

12	Раздел новостей	20	30	40
13	Темы оформления	10	15	20
14	Мобильная адаптация	30	40 (осталось 14 часов)	50
15	Раздел «Контакты»	10	15	20
16	Раздел добавления (автора, тайтла, команды)	10	15	20
17	Создание даталогической модели БД	10	12	14
18	Настройка git репозитория	6	8	10
19	Проектирование контрактов API	15	20	25
20	Базовая настройка окружения и подключение зависимостей (docker, frontend, backend)	8	10	12
21	Выбор стека технологий	12	15	18
22	Модульное тестирование	70	80 (30 часов осталось)	90
23	Интеграционное тестирование	110	120	130
24	Приемочное тестирование	50	60	70

По плану на весь проект у нас затрачивается **720** человеко-часов. Прошло уже 75 процентов срока, но фактически выполнена только половина задач. Они выделены зеленым. Суммарно это **341** человеко-часов. Осталось **720-341 = 379** чч работы. Нам нужно отказаться от части запланированных задач, иначе мы не успеем в срок. Выделим их красным. В эту категорию войдут не такие важные и критические для пользователя задачи.

- Социальная составляющая - просмотр обсуждений, раздел контактов, новостей. Это важные разделы для вовлечения и удержания аудитории, однако в текущей ситуации лучше предоставить основной функционал в лучшем виде, ведь обсудить прочитанные тайтлы пользователи смогут и в других соц. сетях.
- Редактирование манги - пока сконцентрируемся только на чтении. Редактированием при необходимости первое время будут заниматься администраторы, у пользователей это не такая популярная функция.
- Интеграционное тестирование - слишком трудозатратный процесс, поэтому пока от него откажемся. Оставим приемочное, чтобы убедиться, что основной функционал работает. Как раз пользователи помогут нам на первых порах

обнаружить и пофиксить ошибки, потенциально упущенные на интеграционном тестировании.

Итого суммарно, отказавшись от ряда задач, мы сэкономили **201** человеко-час.
Осталось $379 - 201 = 178$ чч.

379 часов еще нужно потратить по плану на оставшиеся задачи, но у нас осталось 180 часов. Сократим некоторые задачи. Выделим их красным. Вместе с ним сократиться и модульное тестирование тех разделов, которые мы убрали - сэкономили 201 час. 178 часов у нас запланировано на оставшиеся задачи.

Оценка возможностей команды

Наша сетевая диаграмма с учетом убранных задач теперь выглядит вот так, где красными стрелками выделены те разделы, которые еще не реализованы полностью.

В ходе выполнения выяснилось, что фронтовые задачи начали идти быстрее, за счет того, что зачастую они эффективно переиспользуют уже написанные компоненты из других задач и разделов. Поэтому фронтовые части разделов закрываются быстрее бэковых. У нас появляются простаивающие фронтэнд-инженеры, которые не имеют компетенцию в написании бэкенда или в его модульном тестировании. Однако, они могут заняться приемочным тестированием. Таким образом, за счет освободившихся рук, мы сможем повысить параллельность выполнения задач и сократить суммарное время выполнения еще на **60**. Итого у нас останется $275 - 60 = 215$ чч работы. Сделаем сводку по вышеописанным оптимизациям:

Было суммарно: 720

Сделано: 341

Отказались: 201

Осталось делать: 178

Задачи по настройке и внедрению процессов заняли больше времени, чем планировалось, из-за чего возникла такая ситуация с горящими сроками, однако команда за $\frac{3}{4}$ срока сработалась и теперь 178 чч оставшихся задач за $\frac{1}{4}$ срока не будут большим вызовом, как могло показаться.

Вывод

В ходе выполнения данной лабораторной работы мы попробовали справиться с ситуацией, когда команда не укладывается в отведенные сроки, то есть проект не был оценен корректно в начале, либо что-то пошло не по плану. Такое встречается довольно

часто и мы сумели выделить основной функционал, который необходимо добить, чтобы представить готовый продукт пользователям а также оптимизировали время выполнения у части задач. По итогу получился вполне достижимый результат.