**Процессы для поддержания жизненного цикла ПО «Hyper Adtech»**

1. **Введение**

Настоящее руководство описывает процессы, обеспечивающие поддержание

жизненного цикла программного обеспечения «Hyper Adtech».

1. **Описание процессов**

Документация, содержащая описание процессов, обеспечивающих поддержание жизненного цикла программы, в том числе устранение неисправностей, выявленных в ходе эксплуатации программы, совершенствование программы, а также информацию о персонале, необходимом для обеспечения такой поддержки

# **Первоначальная постановка задач**

## **Создание**

Все задачи заводятся в трекере задач. В задаче указываются в свободной форме:

1. Что нужно сделать.
2. Проблема (зачем это нужно сделать).
3. Цель (что принесет эта разработка).

## **Приоритезация**

В начале спринта (по понедельникам, раз в две недели) все задачи оцениваются на планировании, где участвуют менеджеры проектов со стороны разработки, менеджеры продукта и стейкхолдеры задач. Задачи оцениваются по степени важности для бизнеса и потенциальной прибыли, часть наиболее важных задач берется в работу на 2 недели, остальные откладываются до следующего планирования.

## **Функциональное описание**

На первом этапе задача описывается функционально: определяется ключевая проблема, которую хотим решить, составляется перечень гипотез, выбирается способ решения. Создается страница в проекте продукта с описанием экранов, примерным функционалом и описанием эксперимента при внедрении. Далее задача оформляется по шаблону:

Проблема, которую решаем:

Гипотеза 1:

Гипотеза Х:

Типы пользователей: [не авторизованный/авторизованный, без подписки/триал/подписка]

Типы профилей: [взрослый/детский]

Описание работы: [ссылка на функциональное описание]

Прототип: [ссылка на Sketch Cloud]

Дизайн: [ссылка на Zeplin]

Коммуникация: [описание коммуникаций с пользователем]

## **Прототипирование**

На базе библиотеки элементов создается предварительный интерактивный прототип интерфейса. Прототип обсуждается со стейкхолдером. Перед передачей в дизайн прототип и функциональное описание проверяется по чек-листу:

1. Описана механика работы;
2. В прототипе учтены все типы пользователей и типы профилей.

## **Дизайн**

На основе прототипа и функционального описания дизайнер создает поэкранный дизайн интерфейса. Перед передачей в разработку дизайн проверяется по чек-листу:

1. Есть макеты для всех типов устройств;
2. Макеты содержат все необходимые состояния, согласно механике работы;
3. В макетах учтены все типы пользователей и типы профилей;
4. Макеты сделаны с учетом адаптива под основные разрешения экрана.

## **Написание текстов**

Копирайтер пишет/правит тексты по прототипу

## **Ревью разработки**

Функциональное описание дополняется ивентами, методами и тегами эксперимента. Оценивается срок разработки.

## **Разработка**

Разработка

## **Ревью продукта**

После окончания разработки и тестирования задача проходит ревью продукта. На этом этапе оценивается соответствие разработанного продукта изначально поставленной задаче.

# **Процесса разработки и поддержания жизненного цикла программ**

1. Процесс постановки задач в разработку и действия с ними
2. Информация о встречах и демонстрациях

## **Типы задач:**

* Эпик. Глобальная задача, представляет собой основную user story, связанная с задачей из раздела первоначальной постановки задач.
* Разработка. Задача по разработке нового функционала, привязанная к определенному эпику.
* Рефакторинг. Доработка существующей функциональности продукта.
* Ошибка. Проблема, найденная в текущем функционале.

Для каждого проекта ведется своя релизная история в формате A.B.C начиная с 1.0.0, где:

* A – главный номер версии (major version number)
* B – вспомогательный номер версии (minor version number)
* C – номер итерации по работе над функционалом версии A.B

Статусы и переходы между задачами применимы ко всем проектам разработки.

Для формирования общей очереди разработки используется доска типа Kanban. Приоритет задачи выставляется с учетом местоположения в списке (чем выше, тем приоритетнее) и выставляется менеджером проекта для каждого разработчика.

Разработка ведется недельными спринтами, спринты формируются в конце предыдущей недели менеджерами проекта.

## **Тип задачи Эпик.**

### **Предварительная оценка.**

Эпик создан и назначена на тимлида. По тому описанию, что есть у эпика, в течении двух дней необходимо дать предварительную оценку по нему. Никаких работ по эпику в данном статусе не проводится. После получения предварительной оценки эпик переводится на группу менеджеров проекта.

### **Оценка.**

В данном статусе эпик имеет ТЗ с полным описанием, необходимым разработчику для выполнения задачи. Эпик назначается на тимлида (или на конкретного разработчика) для оценки и должен быть оценен в течении недели. В случае, если у эпика неполное\непонятное описание, эпик возвращается на менеджера проекта с необходимыми комментариями.

Эпик декомпозируется на задачи и возвращается на менеджера проекта. Декомпозированные задачи назначаются на разработчиков, которые будут непосредственно выполнять данную задачу.

### **В работе.**

После начала работы Эпик переводится в данный статус.

### **Закрыта.**

После окончательного релиза эпик закрывается менеджером проекта.

## **Тип задач Разработка, Рефакторинг и Ошибка.**

### **Оценка.**

Задача в данном статусе имеет подробное описание и назначается на разработчика, который будет непосредственно выполнять задачу. В случае, если описание неполное/непонятное, их нужно устранить на данном этапе. Оценка должна быть предоставлена разработчиком в течении двух дней, после чего переведена на группу менеджеров проекта.

### **Ожидает выполнения.**

Задачи в данном статусе формируют общую очередь разработки, имеют исполнителя и четкие сроки выполнения. Задачи берутся разработчиками в работу согласно приоритету на доске.

### **В работе.**

Задача переводится в данный статус разработчиком в момент начала работы над задачей.

### **Ревью.**

После выполнения задачи она переводится в статус ревью и назначается на какого-либо разработчика. В случае, если были найдены ошибки в задаче или не выполнен функционал, описанный в задаче, задача переводится в статус “В работе” и назначается на ответственного разработчика.

### **Ожидает тестирования.**

Задача переводится в данный статус после ревью и деплоя на тестовый сервер и назначается на тестировщика.

### **Тестирование.**

Задача переводится в данный статус тестировщиком в момент начала работы над задачей. В случае, если были найдены ошибки в задаче или не выполнен функционал, описанный в задаче, задача переводится в статус “В работе” и назначается на ответственного разработчика.

### **Протестировано.**

После успешного тестирования задача переходит в данный статус и назначается на менеджера проекта.

### **Ожидает релиза.**

В данном статусе из готовых задач формируются релизы. После формирования релиза создается задача для копирайтера для написания текстов релизов. Все релизы (и задачи, которые в него входят) фиксируются в сервисе документации. Не все задачи, которые находятся в данном статусе, уходят в ближайший релиз.

### **Релиз.**

После релиза задачи переводятся в данный статус.

### **Закрыто.**

Задачи закрываются после проверки на бою. В случае, если были найдены ошибки, то задачи возвращаются в статус “В работе”.

### **Отменена.**

Задача может быть отменена в любом статусе с обязательным указанием причины отмены.

## **Тестирование и релиз.**

В рамках разработки используется несколько серверов: DEV, STAGE, PROD.

Тестирование должно проводиться на всех серверах.

1. DEV сервер.

После того, как задача выполнена разработчиком и произведен ревью кода, делается мр на дев сервер. После успешного тестирования на дев сервере задача деплоится на стейдж сервер.

2. STAGE сервер.

Задачи собираются на стейдж сервере, где проводится дополнительное тестирование. В гитлабе формируется релиз, куда попадают только те задачи, которые должны уйти в прод.

3. PROD сервер.

Проводится финальное тестирование, после чего задачи закрываются.

## **Собрания и встречи.**

### **Встречи с командой разработки.**

Раз в неделю (в конце каждой недели) проводятся общие 20-ти минутные встречи между менеджерами проекта и командой разработки для подведения итогов текущего спринта и составления планов на следующий спринт.

### **Общие встречи.**

Раз в неделю необходимо проводить общие встречи между менеджером продукта и менеджерами проекта, на которых будут подводиться промежуточные итоги реализации глобальных задач.

Перед релизом проводится общая встреча, на которой проводится итоговая демонстрация релиза. Результатом демонстрации должно быть решение по релизу функционала либо отправка на доработку.

В случае, если релиз небольшой (хотфиксы, мелкие правки и доработки), по согласованию сторон релизную встречу можно не проводить.

Любые другие встречи проводятся по согласованию сторон в любое удобное время.