

Веб-портал разработки ПО

Описание, спецификация

1. Описание системы

WellspringSocial представляет собой инновационную гибридную систему, целью которой является создание кроссплатформенных мобильных приложений при участии распределенной группы разработчиков с использованием уникальной системы, максимально автоматизирующей производственный процесс.

К основным составляющим WellspringSocial относятся:

1. Система, позволяющая максимально автоматизировано создавать кроссплатформенные мобильные приложения;
2. Социальная среда, позволяющая объединять разработчиков, эффективно распределяя и разделяя между ними задачи.

1.1. Структура социальной среды разработчиков

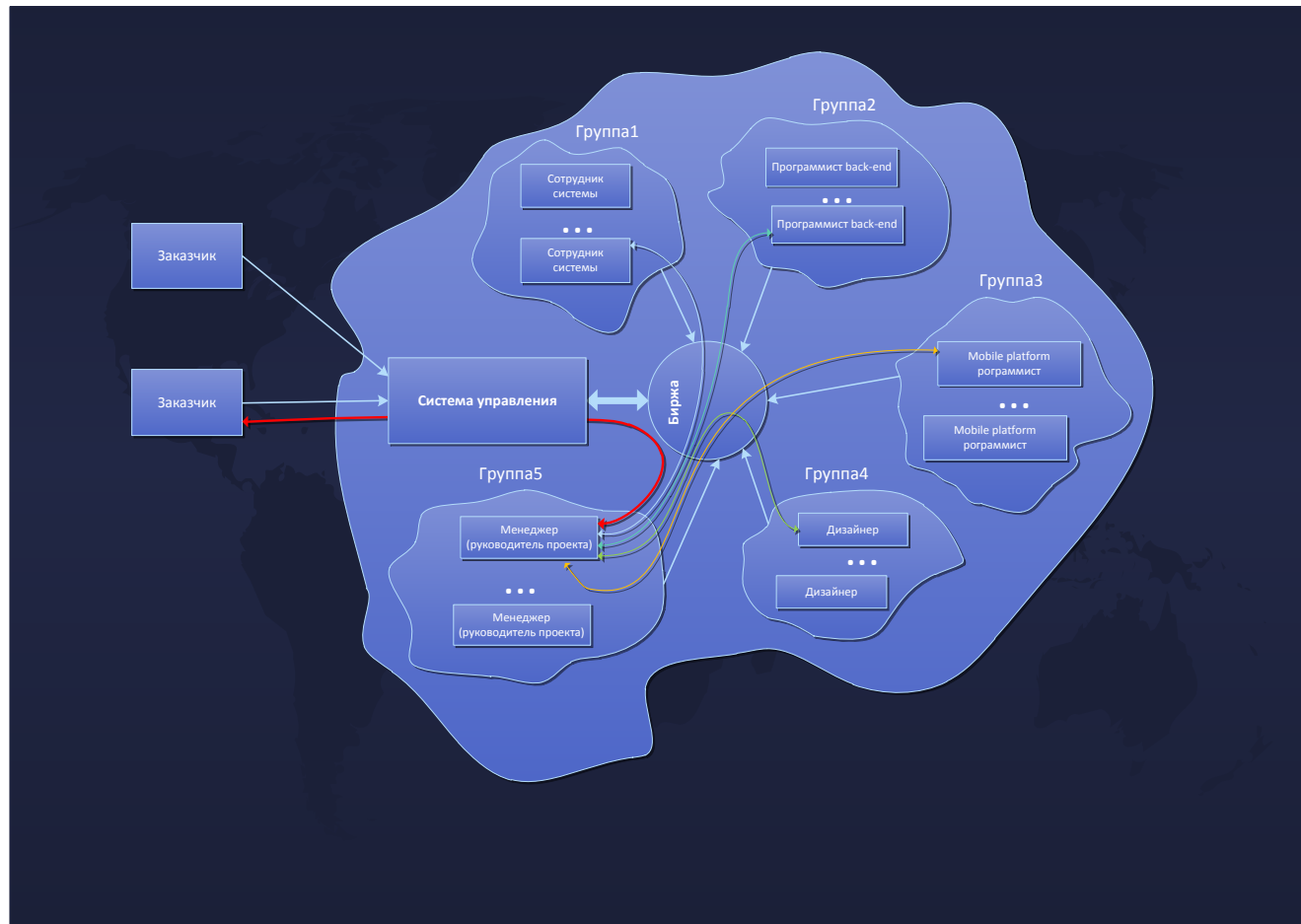


Рис. 1. Общая структура взаимодействия участников социального сообщества проекта WellspringSocial

На рисунке 1 показана схема взаимодействия основных участников процессов разработки кроссплатформенного мобильного приложения, к которым относятся:

а) Заказчик;

Является юридическим, или частным лицом, которому необходимо создание качественного мобильного приложения в кратчайшие сроки.

б) Руководитель проекта (менеджер проекта);

Активный участник социального сообщества системы WellspringSocial. Выполняет следующие функции:

- Координация и корректировка действий исполнителей;
- Создание концепта приложения на основе предпочтений, требований, и желаний заказчика;
- Выбор исполнителей;
- Распределение задач между исполнителями;
- Контроль сроков и качества работ исполнителей;
- Согласование документов по этапам разработки;
- Иные организационные задачи.

в) Программист (для мобильных платформ)

Активный участник социального сообщества WellspringSocial. Выполняет следующие функции:

- выполняет мониторинг биржи работ;
- участвует в конкурсе на принятие работы, находящейся в стеке заданий биржи;
- выполняет поставленную задачу;
- участвует в отладке и тестировании приложения и собственного компонента;
- имеет возможность поиска других разработчиков компонента (другие платформы, распределение сложной задачи);
- не имеет возможности общения с заказчиком напрямую.

г) Программист (сервисы получения/передачи данных)

Активный участник социального сообщества WellspringSocial. Выполняет следующие функции:

- выполняет мониторинг биржи работ;
- участвует в конкурсе на принятие работы, находящейся в стеке заданий биржи;
- выполняет поставленную задачу при помощи встроенного в виртуальное рабочее пространство WellspringSocial инструмента создания API. На момент начала разработки методов API автоматически создаются заглушки на методы API;
- имеет возможность поиска других разработчиков компонента (другие платформы, распределение сложной задачи);
- не имеет возможности общения с заказчиком напрямую.

д) Дизайнер

Активный участник социального сообщества WellspringSocial. Выполняет следующие функции:

- выполняет мониторинг биржи работ;
- участвует в конкурсе на принятие работы, находящейся в стеке заданий биржи;
- выполняет поставленную задачу;
- производит верстку готового приложения при помощи встроенного в виртуальное рабочее пространство WellspringSocial графического редактора;
- имеет возможность поиска других разработчиков компонента (другие платформы, распределение сложной задачи);
- не имеет возможности общения с заказчиком напрямую.

е) Сотрудник системы

Может быть в штате проекта WellspringSocial, или удаленным пользователем системы, обладающим необходимыми правами и высоким уровнем доверия. Выполняет следующие функции:

- контроль соблюдения заказчиком и исполнителем договорных обязательств;
- решение спорных ситуаций между любыми звеньями системы;
- контроль соблюдения всеми звеньями системы правил, описанных в регламентных документах WellspringSocial;

- контроль исполнения взаиморасчетных обязательств.

Структура социальной среды WellspringSocial (рис.1) имеет следующие особенности:

- исполнители не имеют возможности общения напрямую. Взаимодействие между исполнителями производится через «Биржу», тем самым исключая утечки информации по проекту, а также позволяет вести контроль со стороны системы над уровнем загруженности участников социальной среды;

- система управления представляет собой виртуальное рабочее пространство, предоставляющее следующие возможности:

- а) функционал предоставляется с разделением по ролям участников социального сообщества WellspringSocial;
- б) предоставление функционала системы контроля над исполнением задач по проекту (ERP система);
- в) предоставление функционала управления отчетностью;
- г) предоставление функционала, необходимого для осуществления процедур подписания договорных документов и актов;
- д) предоставления функционала, необходимого для осуществления платежных операций;
- е) предоставление функционала работы с «Биржей» заявок;
- ж) предоставление информации о настраиваемых и пользовательских компонентах приложения в ComponentsMarket;
- з) предоставление функционала, необходимого для осуществления общения между участниками проекта, поддерживающим: мультязычный чат, голосовое общение, интерактивный механизм согласования задач с подробным сохранением действий;
- и) другое.

- заказчик не имеет доступа к возможностям «Биржи»;

- заказчик имеет право общаться напрямую только с руководителем проекта (менеджер);

- в момент урегулирования спорных моментов заказчик имеет возможность общения с конечным исполнителем в общем чате с участием руководителя проекта;

- поддерживает механизмы получения и загрузки пользовательских модулей;
- все участники процесса разработки мобильного приложения несут ответственность в соответствии со своими ролями;
- урегулирование спорных ситуаций, контроль исполнения процедур оплаты, ведет сотрудник проекта WellspringSocial;

1.2. Основные этапы разработки мобильных приложений

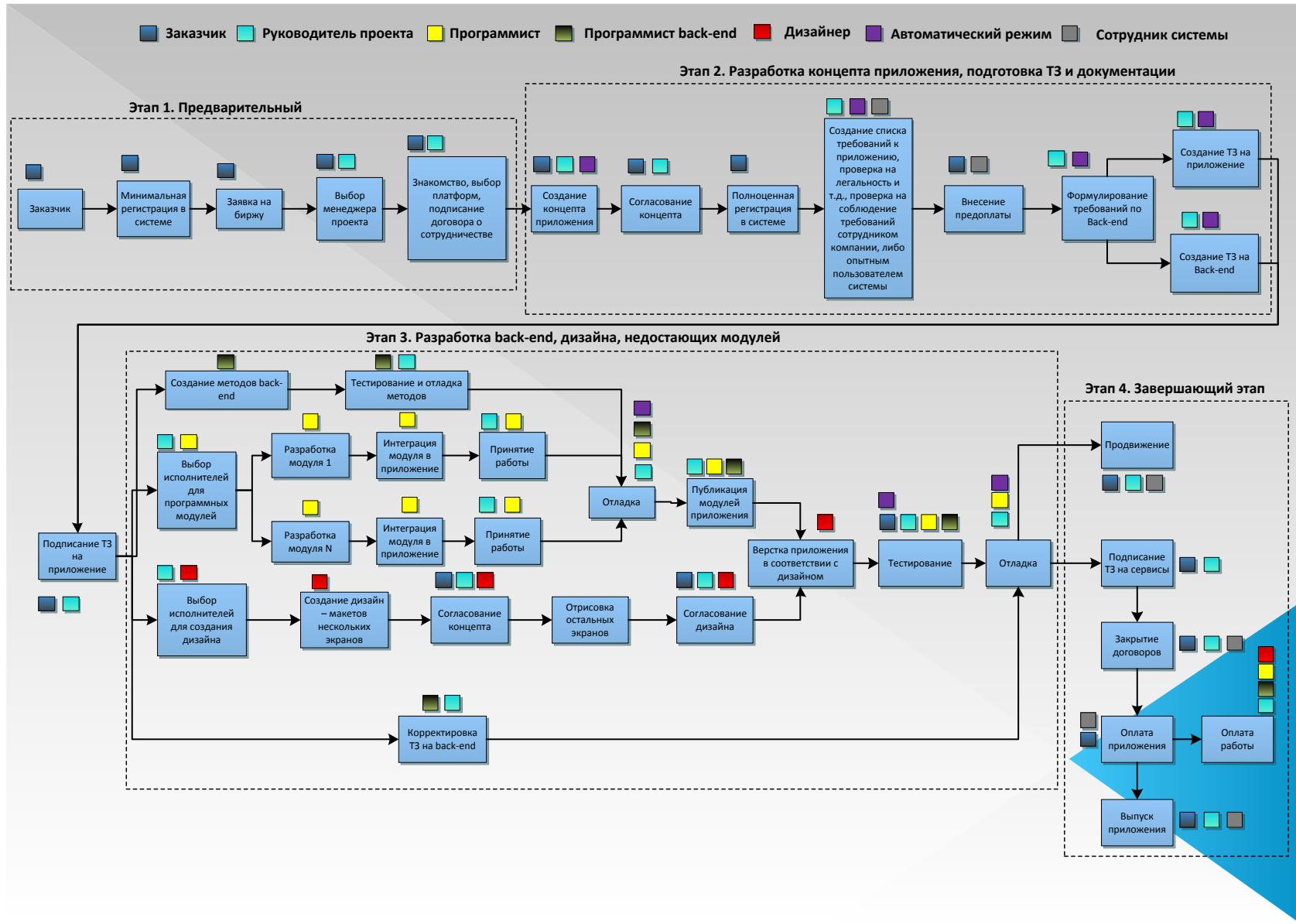


Рис. 2. Этапы разработки мобильного приложения в системе WellspringSocial

На рисунке 2 показаны этапы разработки мобильного приложения в социальной среде WellspringSocial.

1.2.1. Предварительный этап

а) минимальная регистрация в системе

На данном этапе потенциальный заказчик производит минимальную регистрацию в системе WellspringSocial, если он не был зарегистрирован.

К обязательным данным относятся: имя пользователя, пароль, адрес электронной почты, ФИО, дата рождения, язык, место проживания.

б) заявка на «Биржу»

После получения доступа к собственному рабочему столу, заказчик может сформировать заявку на разработку мобильного приложения, которая после оформления отправляется на «Биржу». В режиме очереди на нее могут откликнуться участники социальной среды WellspringSocial, имеющие права руководителя проекта (менеджера).

в) выбор менеджера проекта

В том случае, если заявок на выполнение работ со стороны руководителей проектов несколько, заказчик имеет право выбрать необходимого менеджера на основе:

- рейтинга руководителя проекта;
- количества и качества выполненных проектов;
- наличия опытной команды разработчиков;
- личного собеседования.

г) знакомство, выбор мобильных платформ, подписание договора о сотрудничестве

После выбора исполнителя, заказчик подписывает с ним договор о сотрудничестве, в котором содержится краткая информация о проекте, а также производится выбор инструментов для реализации.

1.2.2. Разработка концепта приложения, подготовка ТЗ и документации

а) создание концепта приложения

На этом этапе заказчик и руководитель проекта детально описывают приложение и его функционал при помощи конструктора приложения (в дальнейшем - «Конструктор»). Конструктор представляет собой встроенный инструмент в систему управления WellspringSocial, позволяющий визуально отображать экраны и компоненты мобильного приложения, а также строить между ними функциональные связи.

Полученный концепт приложения можно запустить на интегрированном в систему управления WellspringSocial эмуляторе устройства для осуществления визуальной оценки функциональности мобильного приложения.

Стоит отметить, что на данном этапе разработки инструмент разработки «Конструктор» не должен иметь расширенной настройки компонентов и взаимосвязей между ними, так как это обуславливается невысоким уровнем знаний в области программирования у руководителя проекта и заказчика.

б) согласование концепта

На данном этапе достигается договоренность по построенному концепту приложения при помощи «Конструктора» с детальным описанием функционала, которая в дальнейшем не подлежит изменениям.

в) полноценная регистрация в системе

На этом этапе Заказчик обязан сообщить о себе детальную информацию, необходимую для составления юридических документов, а также проведения платежных операций.

г) создание списка требований к приложению, проверка на легальность, и т.д.

Создание списка требований к приложению необходимо для дальнейшего осуществления процедуры автоматической классификации задач по ролям конечных исполнителей, а также необходимости предобработки данных для системы автоматической генерации технических заданий.

Также на этом этапе происходит проверка на легальность будущего контента приложения, заказывается экспертная оценка возможности реализации приложения в соответствии с согласованным концептом приложения.

д) внесение предоплаты

Внесение предоплаты необходимо для оплаты работы конечных исполнителей, к которым относятся:

- программист мобильной платформы. Работа оплачивается полностью.

- программист back-end. Работа оплачивается полностью.

- дизайнер. Создание эскизов приложения не оплачивается. В случае утверждения эскиза приложения, заказчик вносит на счет проекта 50% от стоимости дизайна по утвержденному эскизу.

Замечание: оплата работы менеджера проекта осуществляется после полной сдачи проекта.

Пояснение:

На рисунке 3 показан механизм оплаты работы исполнителей, который сводится к следующим правилам:

- 1) Любую платежную операцию заказчик производит на счет WellspringSocial;
- 2) Руководитель проекта оплату за посреднические услуги получает после полной сдачи проекта;
- 3) Любой исполнитель имеет собственный счет в социальной среде WellspringSocial;
- 4) Счет проекта представляет собой инструмент распределения денежных средств проекта по счетам исполнителей;
- 5) Предоплата Заказчика должна распределяться между конечными исполнителями таким образом, чтобы в случае невозможности продления контракта на разработку мобильного приложения со стороны Заказчика, оплата работ по разработке пользовательских программных компонентов была произведена полностью.
- 6) Снятие средств со счетов конечных исполнителей возможно только после полного завершения работ по проекту.

е) формирование требований по back-end

На данном этапе в автоматическом режиме создается перечень методов API, необходимых для работы приложения в соответствии с согласованным концептом приложения.

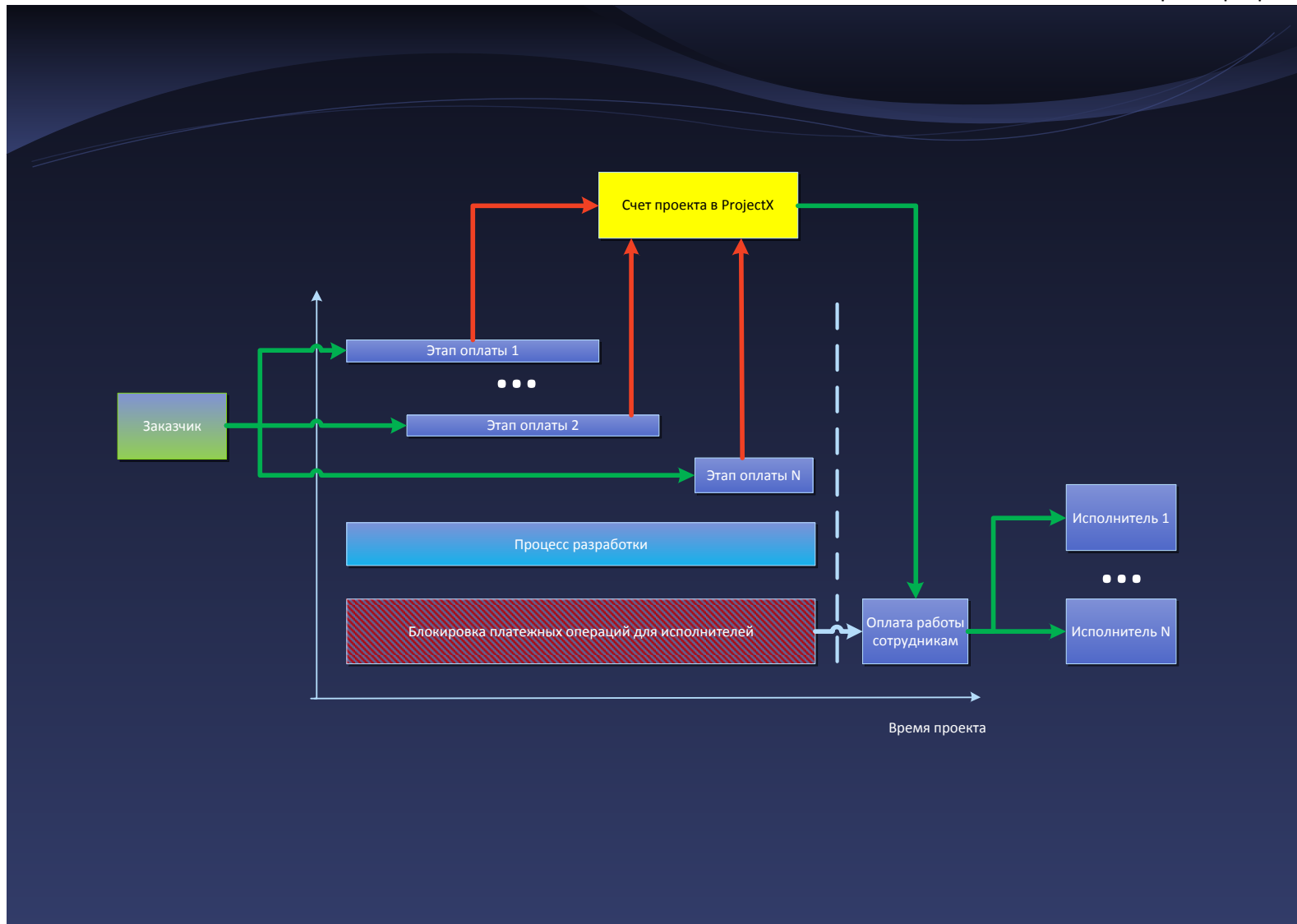


Рис. 3. Процедура оплаты выполняемых работ в социальном сообществе WellspringSocial

ж) формирование технического задания на мобильные приложения и back-end

На основе объектной модели компонентов мобильного приложения, полученной при помощи Конструктора приложений, формируется список задач и требований, необходимых для формирования технического задания. На данном этапе формирование технического задания

на разработку мобильного приложения (для необходимых мобильных платформ) производится полностью в автоматическом режиме, учитывая детализированное описание каждого компонента. Такой подход имеет ряд существенных преимуществ перед неавтоматизированным методом формирования технического задания, в котором всегда присутствует вероятность упущения важных, но малозаметных деталей.

з) подписание технических заданий

На данном этапе происходит окончательное согласование документов, содержащих исчерпывающее описание мобильного приложения, которое необходимо разработать. Все последующие требования по изменениям в функционале приложения со стороны Заказчика, оформляются в виде списка дополнительных работ, которые оплачиваются сверх суммы, необходимой для разработки основного функционала.

Замечание:

Процесс исключения ненужного функционала из приложения на этапе разработке сводится к следующим правилам:

- на собственное усмотрение руководитель проекта вправе внести изменения в концепт (и последующее техническое задание) приложения, но при условии, что изменения не понесут существенного усложнения отдельных компонентов приложения, а также суммарный процент изменений не должен превышать 10%;
- в том случае, если изменение функционала приложения существенно в плане реализации, или вероятного увеличения сроков разработки, то подобные работы следует относить к дополнительным.

1.2.3. Разработка back-end, дизайна, недостающих модулей

Этап непосредственной программной реализаций недостающих пользовательских модулей, верстка экранных форм и всех компонентов мобильного приложения, а также полуавтоматическая генерация методов API – самый трудоемкий и продолжительный.

На данном этапе параллельно выполняются следующие задачи:

- **на основе подписанного технического задания формируется список пользовательских компонентов, отсутствующих в WellspringSocial ComponentsMarket.**

1.2.3.1. Реализация пользовательских компонентов:

а) выбор исполнителя для пользовательского компонента

Руководитель проекта должен в кратчайшие сроки в полуавтоматическом режиме отправить заявки по разработке пользовательских компонентов на Биржу работ (Биржа работ отличается от Биржи сотрудников).

Откликнувшиеся на заявку конечные исполнители, к которым относятся программисты mobile-platform, проходят профессиональный отбор менеджером проекта. Основания для выбора исполнителя должны быть следующие:

- 1) Рейтинг исполнителя, являющийся составным показателем эффективности его работы;
- 2) Опыт работы исполнителя;
- 3) Список выполненных работ, а также отзывы руководителей проектов, в которых этот исполнитель принимал участие;

4) Другое.

б) реализация пользовательского компонента

За выбором исполнителя для конкретного пользовательского компонента, следуют действия по:

- заключению договора между руководителем проекта и исполнителем средствами социального сообщества WellspringSocial;
- переводу со счета проекта на счет исполнителя четко определенной суммы за работу над пользовательским компонентом;
- реализации пользовательского компонента.

Замечание: следует отметить, что требования кроссплатформенности могут существенно увеличить количество исполнителей пользовательского компонента приложения (см. Рис.4). Упростить работу руководителя по поиску исполнителей можно при помощи следующего алгоритма:

- 1)** руководитель проекта выбирает необходимый пользовательский компонент, который необходимо разработать для нескольких мобильных платформ, принципиально отличающихся друг от друга;
- 2)** задача по созданию пользовательского компонента с подробным описанием помещается на Биржу компонентов;
- 3)** спустя некоторое время на этот запрос откликается конечный исполнитель **И1**, специализирующийся на разработке мобильных компонентов для одной определенной мобильной платформы;
- 4)** для того, чтобы руководитель проекта не искал конечных исполнителей пользовательского компонента для всех необходимых мобильных платформ в отдельности, исполнитель И1 может взять эту функцию на себя, получив на время разработки расширенные права в социальной среде WellspringSocial;

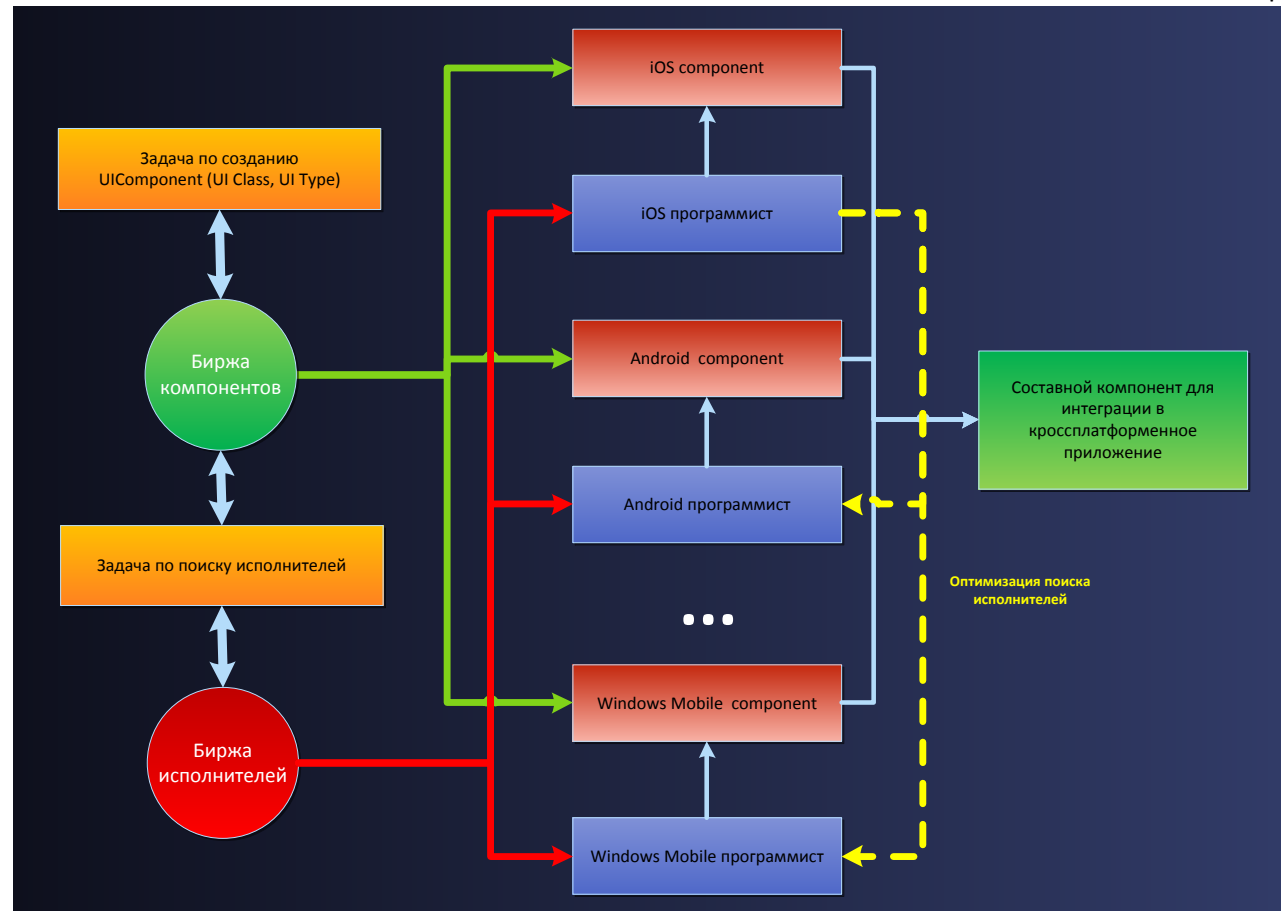


Рис. 4. Процедура создания пользовательских компонентов

в) интеграция пользовательского компонента в мобильное приложение

Разработанные пользовательские компоненты интегрируются исполнителями в кроссплатформенное мобильное приложение при помощи расширенного Конструктора (Рис.5).

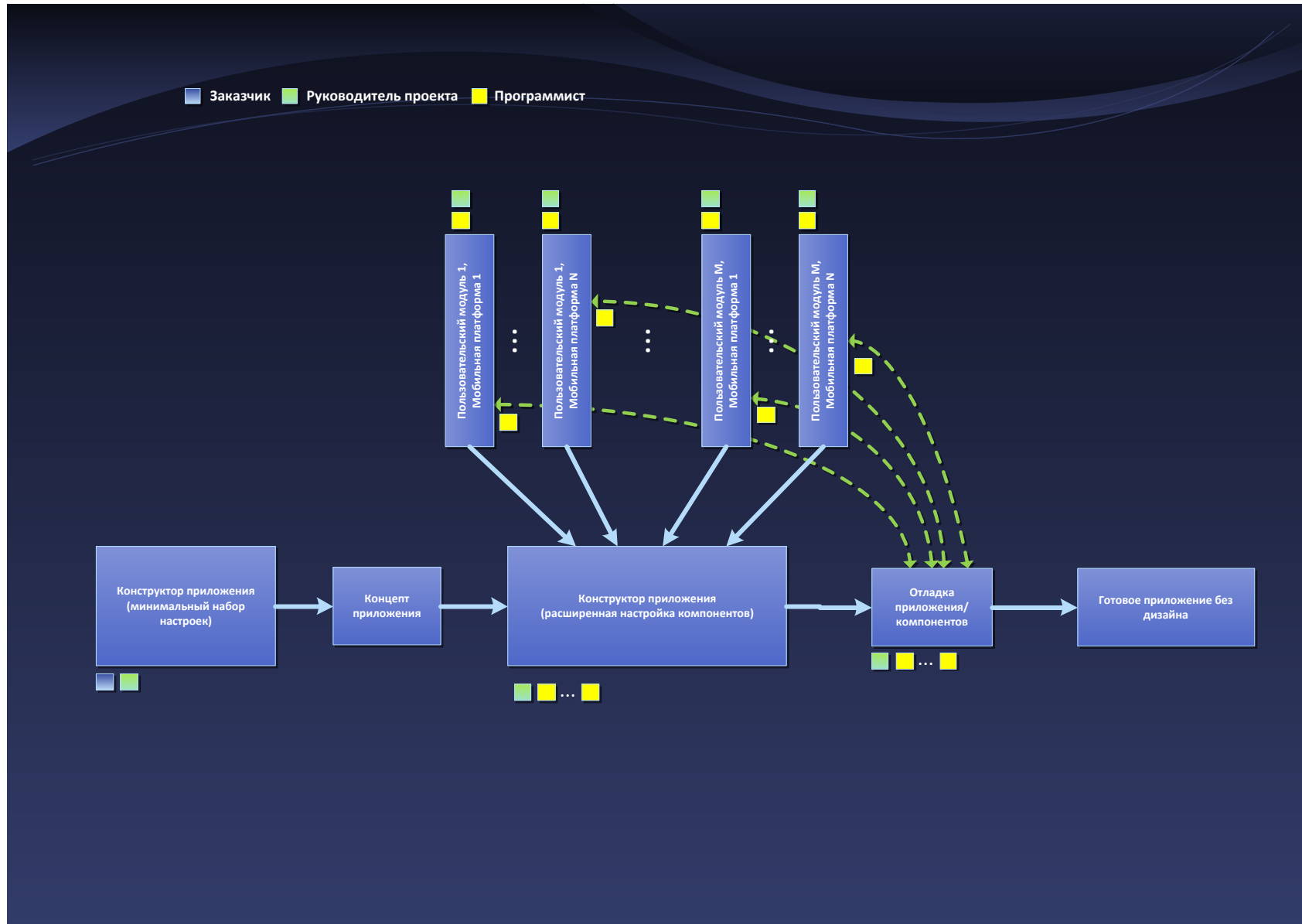


Рис. 5. Механизм интеграции пользовательских компонентов в приложение

г) принятие работы по разработке пользовательского компонента, отладка, отправка компонента в ComponentsMarket

После завершения разработки пользовательского компонента проверяется соответствие программным и функциональным требованиям. В том случае, если пользовательский модуль реализован корректно, производится комплексная проверка работы приложения с акцентом на отказоустойчивость этого модуля.

Отправка пользовательского модуля в ComponentsMarket является необходимым критерием развития социального сообщества WellspringSocial. При формировании заявки для отправки модуля в ComponentsMarket важна следующая информация:

- подробное описание функционала модуля;
- список поддерживаемых мобильных платформ;
- информация о разработчиках модуля;
- информация о мобильных приложениях, в которые этот модуль был интегрирован. Должна поддерживаться функция запуска этих приложения на симуляторах мобильных устройств для проведения оценки качества пользовательского компонента;
- цена модуля;
- отзывы о модуле и его рейтинг;
- описания акции (если она есть), распространяющейся на этого пользовательский компонент;
- другое.

1.2.3.2. Разработка методов back-end

Как говорилось ранее, после согласования концепта приложения в автоматическом режиме формируется список методов API, **необходимых** для работы приложения (Рис. 6).

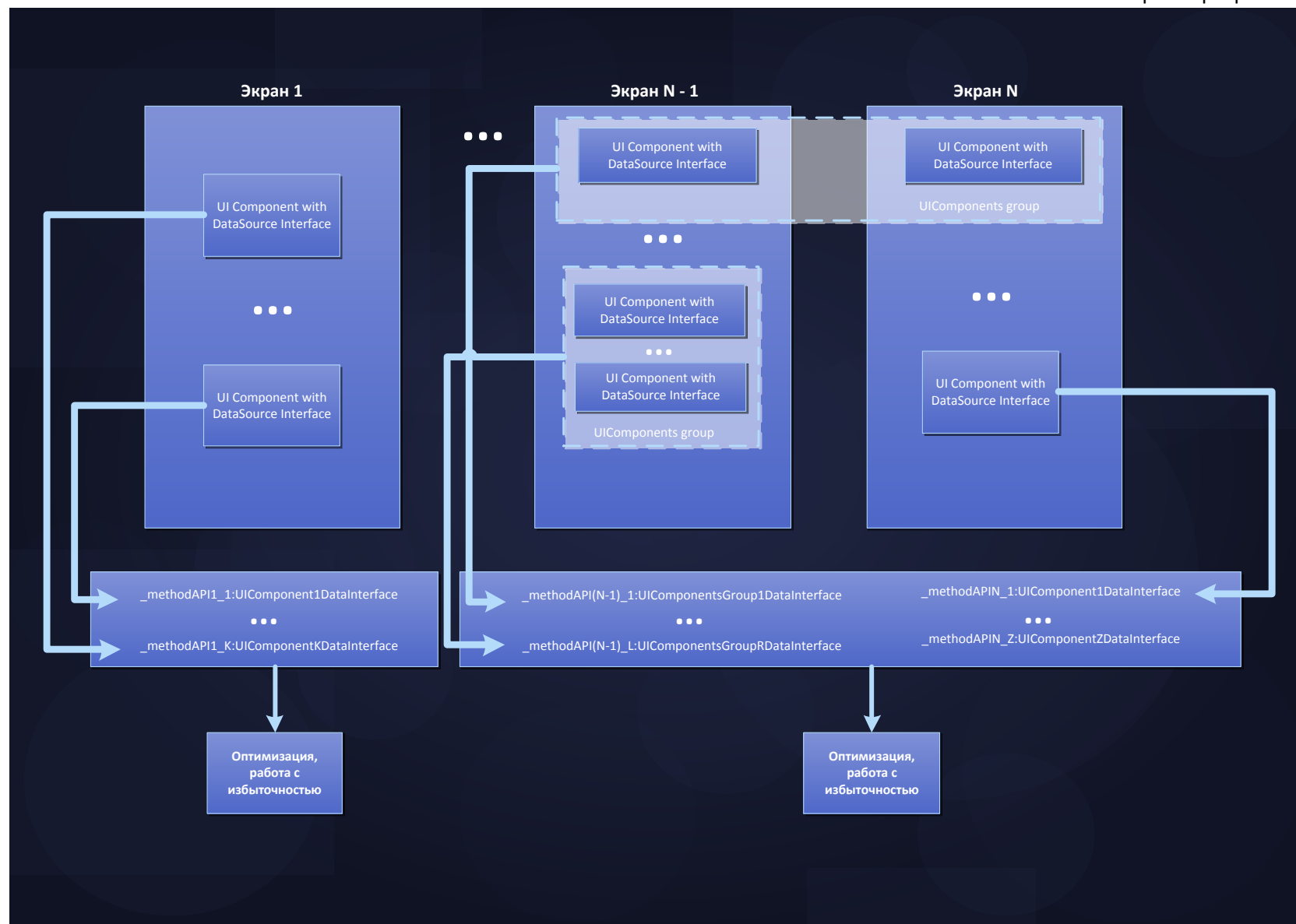


Рис. 6. Механизм получения списка необходимых методов API

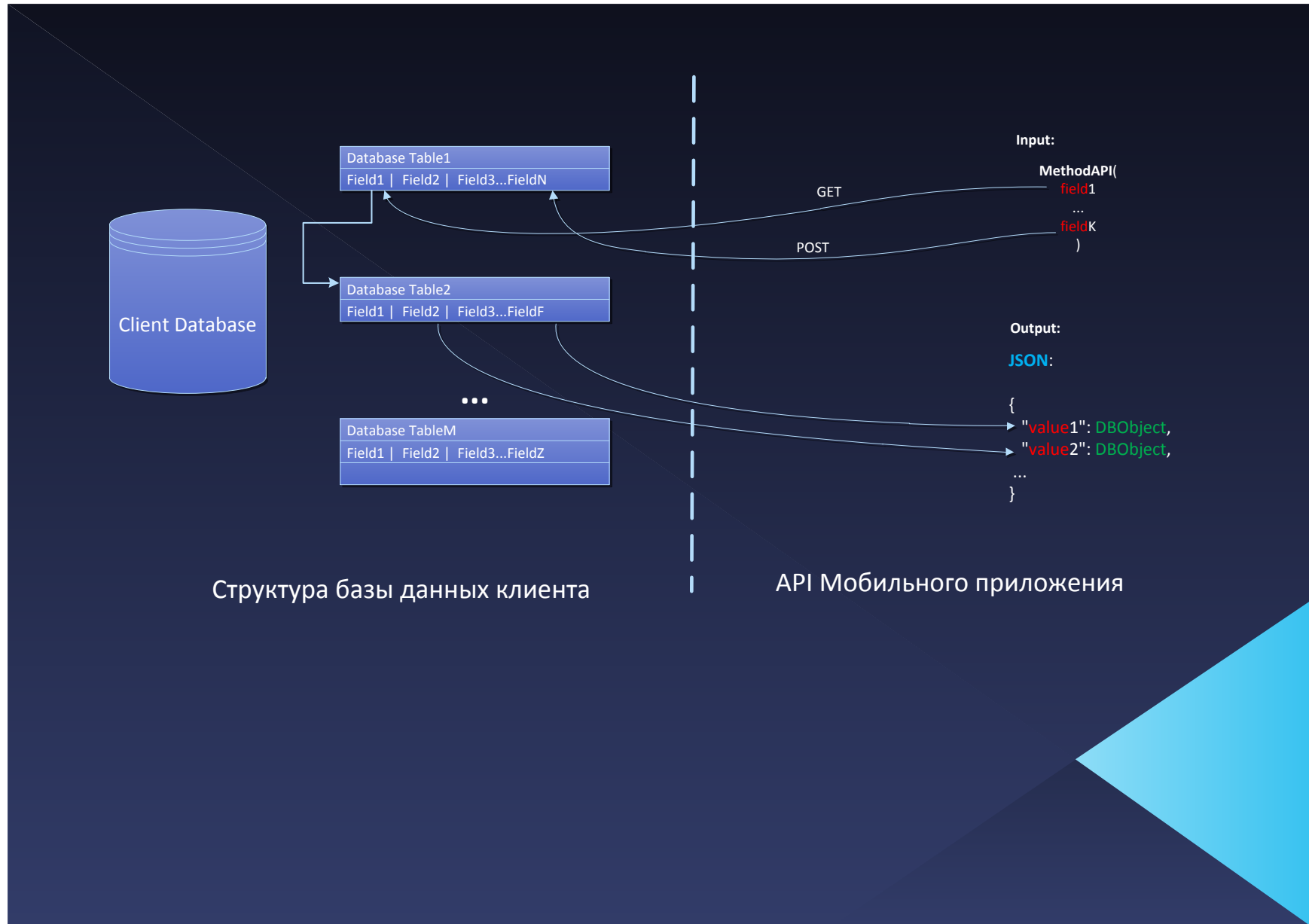


Рис. 7. Метод полуавтоматической генерации API мобильного приложения

На рисунке 7 показан механизм создания методов back-end мобильного приложения.

Действия программиста back-end проекта должны сводиться к следующим действиям:

- детальное определение структуры базы данных клиента (в большинстве случаев построение возможно в автоматическом режиме);
- выбор необходимого метода API из списка необходимых для мобильного приложения;
- определение входных и выходных параметров метода;
- сопоставление входных параметров метода с необходимыми полями таблицы базы данных клиента;
- сопоставление выходных параметров метода с необходимыми полями таблиц базы данных клиента;
- корректировка логики запроса производится при помощи встроенного редактора диаграмм **DBQueryEditor**;
- тестирование метода API при помощи встроенного отладчика back-end методов.

Замечание:

Следует отметить, что во многих случаях, у Заказчика может отсутствовать база данных, либо ее структура засекречена. В этом случае возможно следующее решение:

- в автоматическом режиме формируется структура базы данных, на основе списка требуемых методов API;
- оптимизация структуры базы данных производится в полуавтоматическом режиме;
- корректировка логики запроса производится при помощи встроенного редактора диаграмм **DBQueryEditor**;
- производится тестирование методов API при помощи встроенного отладчика back-end методов;
- заказчику предлагается ручное заполнение построенной базы данных при помощи CMS, либо создание прослойки API по заполнению базы данных мобильного приложения.

Так как требования к API могут меняться на всем протяжении разработки, back-end программист должен иметь возможность участия в доработке структуры базы данных, методов, а также оптимизации запросов в целом.

Принятие и утверждение методов API должно производиться только после сдачи приложения, чтобы исключить вероятность доработки приложения.

1.2.3.3. Работа с дизайном

Действия руководителя проекта по поиску и контролю работ исполнителя по созданию дизайна, а также верстке приложения, в целом идентичны процедуре разработки несуществующих модулей приложения.

Перед руководителем проекта ставятся следующие задачи:

- формирование списка требований

Руководитель проекта должен сформировать список кратких требованиям, согласованных с Заказчиком, который должен включать в себя:

- 1) Брендбук Заказчика, если он есть;
- 2) Список экранов мобильного приложения для выбранной мобильной платформы для создания их концептов;
- 3) Предварительная стоимость выполнения всей работы по дизайну;
- 4) Требования по usability;

Замечание: На этапе создания концепта приложения возможно привлечение специалиста по usability (дизайнер usability). Если это сделано не было (либо не требовалось), то этот факт должен быть указан;

- 5) Другое.

- отправка списка требований на Биржу заявок, выбор исполнителя

Руководитель проекта должен в кратчайшие сроки в полуавтоматическом режиме отправить заявку на создание дизайна на Биржу работ.

Откликнувшиеся на заявку конечные исполнители, к которым относятся дизайнеры, проходят профессиональный отбор менеджером проекта. Основания для выбора исполнителя должны быть следующие:

- 1) Рейтинг исполнителя, являющийся составным показателем эффективности его работы;
- 2) Опыт работы исполнителя;
- 3) Список выполненных работ, а также отзывы руководителей проектов, в которых этот исполнитель принимал участие;
- 4) Другое.

- согласование концепта дизайна приложения

Руководитель проекта и дизайнер договариваются о сроках предоставления нескольких вариантов концепции дизайна для выбранных экранов мобильного приложения под одну мобильную платформу.

Следует отметить, что дизайнер за выполнение этой работы не получает денежного вознаграждения, но он вправе использовать разработанные концепты с целью увеличения собственного портфолио.

В процессе согласования концепта приложения принимают участие: Заказчик, руководитель проекта, дизайнер.

Руководитель проекта имеет возможность аргументированно дать совет Заказчику об использовании того, или иного концепта дизайна, зная технологические особенности, или ограничения мобильного приложения, которое необходимо создать.

В результате выбора из нескольких вариантов концепции дизайна одной, в соответствии с которой будет в дальнейшем разработан весь дизайн приложения, должна быть запущена процедура, позволяющая зафиксировать следующие моменты:

- в случае утверждения понравившейся концепции дизайна, исполнитель получает право на реализацию всего дизайна приложения, а также получает предоплату за работу на собственный счет в социальной среде WellspringSocial;

- вся дальнейшая работа дизайнера должна быть разбита на этапы, чтобы руководитель проекта и Заказчик имели возможность ознакомления и внесения корректировок по мере отрисовки экранов.

- руководитель проекта вправе отказаться от услуг выбранного исполнителя по следующим причинам:

а) потеря связи;

б) существенное отхождение от согласованного концепта дизайна приложения;

в) существенное увеличение сроков работы над дизайном приложения, отхождение от сроков и этапности;

В случае снятия дизайнера с проекта, инициируется процедура по поиску нового исполнителя, сопровождающаяся возвращением предоплаты на общий счет проекта. Также важно заметить, что новый исполнитель должен работать по утвержденному концепту дизайна, делясь долей собственного заработка с предыдущим дизайнером проекта. Новый исполнитель также в праве аргументированно указать слабые стороны предыдущей реализации концепта дизайна, предложив свои. Вопрос по выбору новых концептов дизайна полностью ложится на руководителя проекта, который должен убедить в необходимости изменения концепта Заказчика.

- отрисовка остальных экранов, согласование дизайна

Как отмечалось, работа над дизайном должна быть поэтапной, подотчетной, а также полностью прозрачной для руководителя проекта. Дизайн-решения экранов мобильного приложения должны быть доступны для просмотра, ознакомления, внесения корректировок в Конструкторе мобильного приложения, доступ к которому имеет руководитель проекта и Заказчик.

Результатом этого этапа должно быть согласованное решение Заказчика и руководителя проекта о полном завершении работ по дизайну приложения, подтвержденное средствами электронного документооборота системы WellspringSocial.

- верстка приложения в соответствии с утвержденным дизайном приложения

Важной отличительной особенностью системы WellspringSocial является предоставление дизайнеру широкого набора инструментов для осуществления верстки готового приложения, которые позволят исключить взаимодействие с программистами, а также качественно повысить скорость и эффективность работы.



Рис. 8. Общий вид графического редактора WellspringSocial

На рисунке 8 показан внешний вид графического редактора WellspringSocial, позволяющий:

- производить верстку отдельного компонента мобильного приложения, добавляя ему визуальную уникальность;
- работать с анимацией;
- другое.

Замечание: Организация рабочего пространства данной системы должна быть схожей с распространенными графическими пакетами, позволяя экономить время на обучении исполнителя.

На рисунке 9 показаны основные этапы верстки кроссплатформенного мобильного приложения, которые сводятся к:

- открытию готового мобильного приложения в графическом редакторе WellspringSocial

Открытие проекта готового приложения в графическом редакторе позволяет в автоматическом режиме распределить элементы пользовательского интерфейса по слоям, а также сопоставить каждому экрану отрисованный макет дизайнера с необходимыми пояснениями (если выполняет верстку другой дизайнер) для удобства верстки.

- выбор мобильной платформы

Методы верстки, а также алгоритмы добавления анимации пользовательскому компоненту, существенно отличаются для различных мобильных платформ. По этой причине верстку кроссплатформенного приложения необходимо производить для каждой мобильной платформы в отдельности. Поддержка необходимой платформы достигается благодаря подключению к графическому редактору WellspringSocial модуля поддержки мобильной платформы.

Далее, для каждой из мобильных платформ в отдельности:

- получение списка компонентов мобильного приложения

На данном этапе в автоматическом режиме генерируется список пользовательских компонентов, с которыми необходимо произвести работу по верстке.

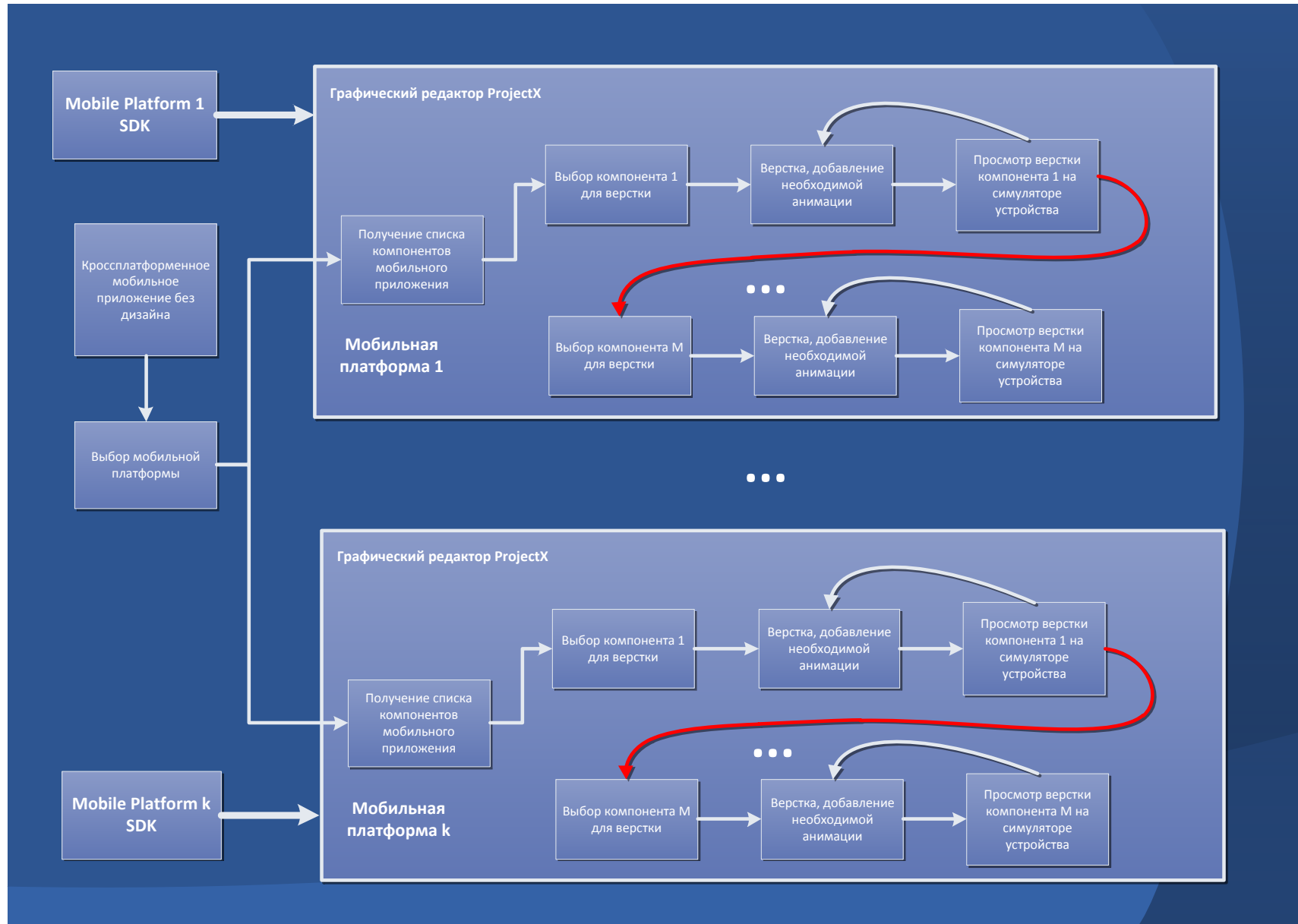


Рис. 9. Этапы верстки мобильного приложения в системе WellspringSocial

- верстка, добавление необходимой анимации

Благодаря графическому редактору системы WellspringSocial производить полную верстку пользовательского компонента, в соответствии с согласованными макетами дизайнера, может один исполнитель. Такой подход дает следующие преимущества:

- а) существенная экономия времени работы по верстке приложения;
- б) количество исполнителей сводится к одному;
- в) верстка приложения, произведенная дизайнером, будет иметь выше качество, нежели произведенная программистом на основе макетов дизайнера;
- г) другое.

- просмотр верстки пользовательского компонента на симуляторе устройства

Система WellspringSocial должна предоставлять возможность запуска приложения на любом этапе верстки для осуществления процедуры визуального контроля корректности выполненных работ.

1.2.4. Тестирование и отладка приложения

Тестирование и отладка приложения является важнейшими этапами в создании качественного и отказоустойчивого приложения.

На основании необходимости тестирования большого количества приложений, выпускаемых одновременно в рамках системы WellspringSocial, очевидным является тот факт, что этот процесс должен быть максимально автоматизирован.

Предлагается следующая автоматизированная схема тестирования мобильного приложения, изображенная на рисунке 10.

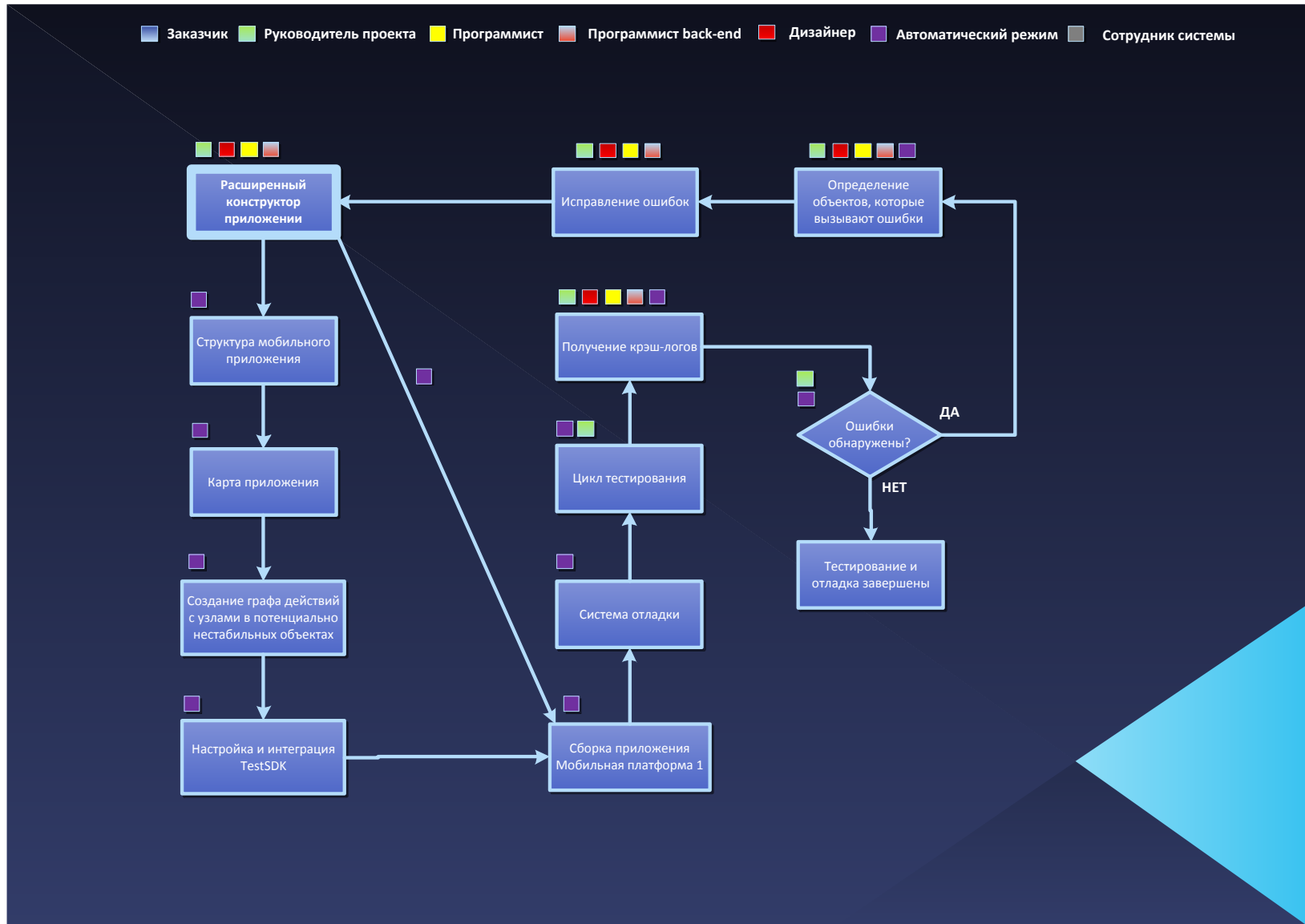


Рис. 10. Тестирование и отладка приложения в системе WellspringSocial

1.2.5. Завершающий этап

1.2.5.1. Подписание технического задания на методы API мобильного приложения

Подписание технического задания на методы получения внешних данных для кроссплатформенного приложения на завершающем этапе обусловлено следующим рядом причин:

- методы API могут меняться на всем протяжении разработки мобильного приложения;
- подписанное техническое задание, содержащее описание методов API, на многих этапах разработки мобильного приложения, может существенно увеличить сроки сдачи проекта;
- после отладки и тестирования приложения формируется список методов API, которые необходимы для стабильной и корректной работы приложения.

Замечание: Подписанное техническое задание на API подразумевает уведомление Заказчика о том, что несогласованное изменение структуры данных по его вине приведет к некорректной работе мобильного приложения;

1.2.5.2. Закрытие договоров

1.2.5.3. Оплата приложения, оплата работ

1.2.5.4. Выпуск приложения

1.2.6. Продвижение

2.0. Описание технической реализации

2.1. Описание системы, выполняющей автоматическую генерацию исходных кодов приложения

Основная идеология системы базируется на общих положениях объектного подхода и включает в себя три основных концепции и три основных этапа развития. Концептуальная схема проекта выглядит следующим образом:

Концепция общих понятий:

- Поскольку все неспецифические языки одним из основных свойств имеют абстрагирование, то существует некое понятийное поле содержащее общие абстрактные понятия.
- Среди языков программирования к языкам с такими свойствами можно отнести объектно-ориентированные языки.
- Для объектно-ориентированных языков полем общих понятий является пересечение понятийных полей.
- Поле общих понятий применяемых в области мобильных приложений охватывает бóльшую часть технологического процесса создания приложений.
- Все общие понятия конкретной группы языков программирования могут быть описаны любым языком программирования содержащим их.

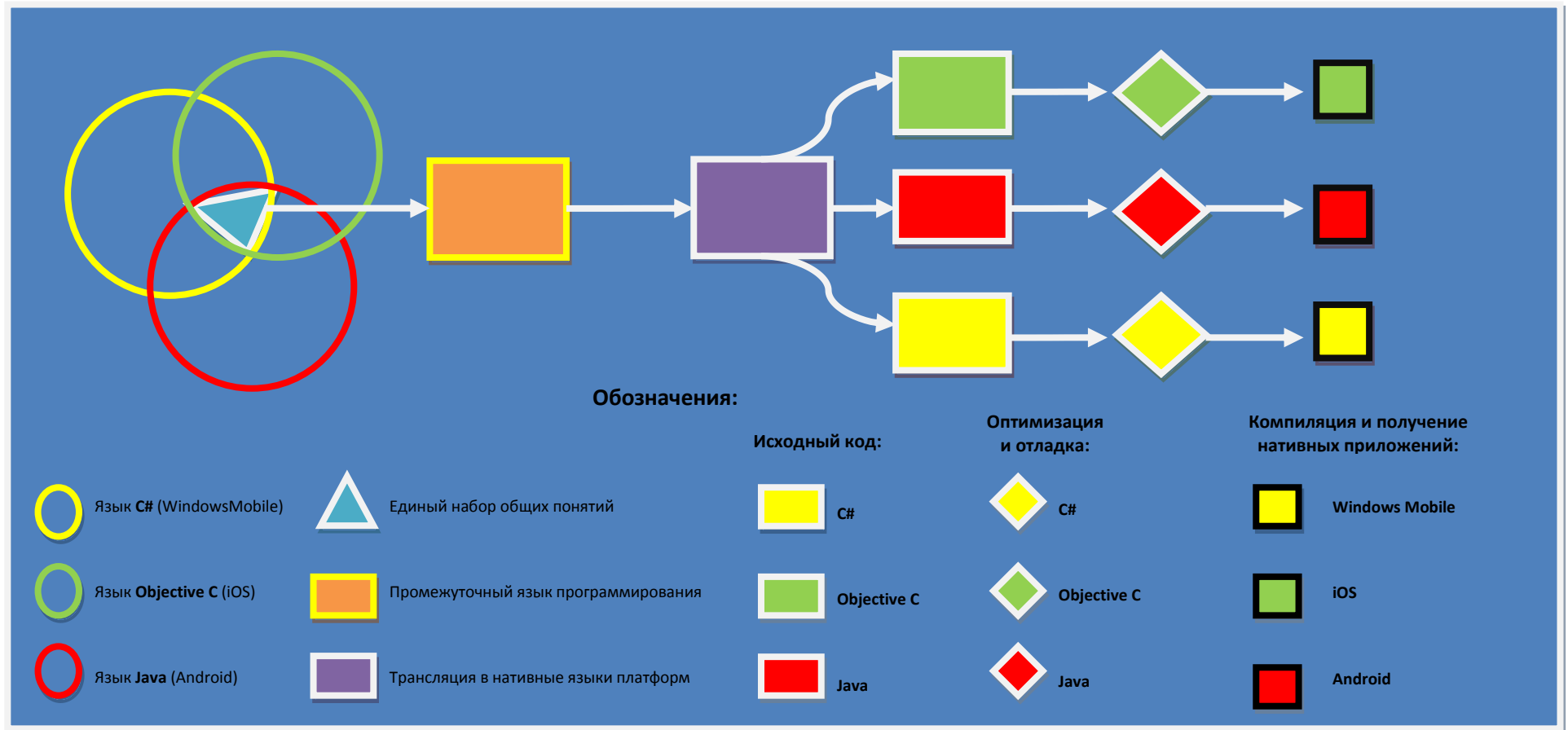
Концепция достаточной декларативности:

- Для создания полного функционала приложения достаточно его описать.
- Для создания полного функционала объекта достаточно его описать.
- Для реализации межобъектных связей и взаимодействий достаточно их описать.
- Для реализации воздействия на объект достаточно описать его реакцию на воздействие.

Концепция остаточной специфичности:

- Любое приложение, создаваемое для работы со специфическим оборудованием, требует специфических приемов оптимизации и отладки.
- Оптимизация и отладка могут быть выполнены на конечном этапе разработки.

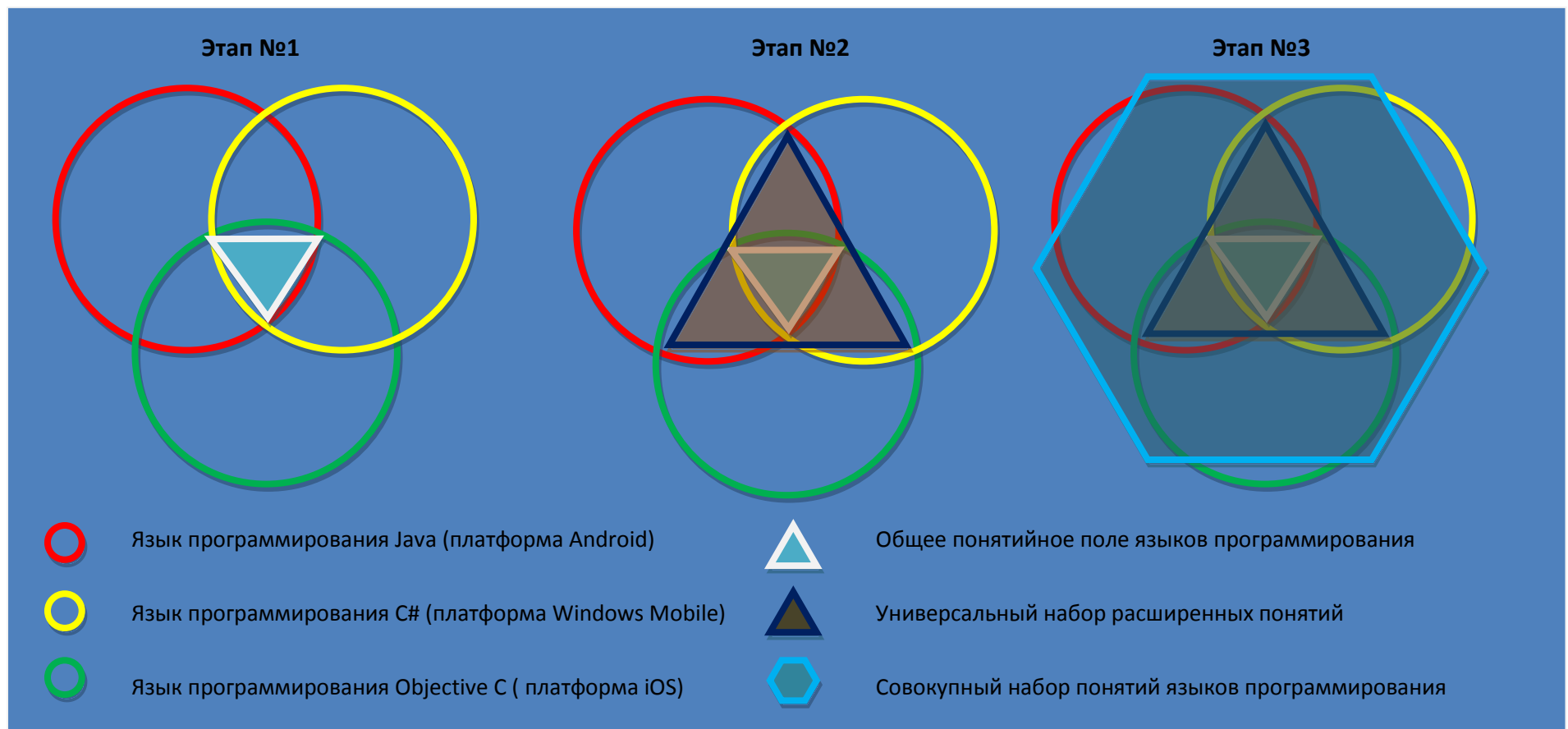
Исходя из вышеуказанных концепций, возможно, разработать такую технологическую схему разработки приложений, при которой универсальность общепонятной базы будет дополняться эффективной оптимизацией и качественной отладкой.



Поскольку единая область понятий, являющаяся пересечением понятийных множеств различных языков программирования, довольно невелика, то предполагается дальнейшее развитие промежуточного языка программирования в сторону расширения при помощи описания специфических понятий при помощи общих.

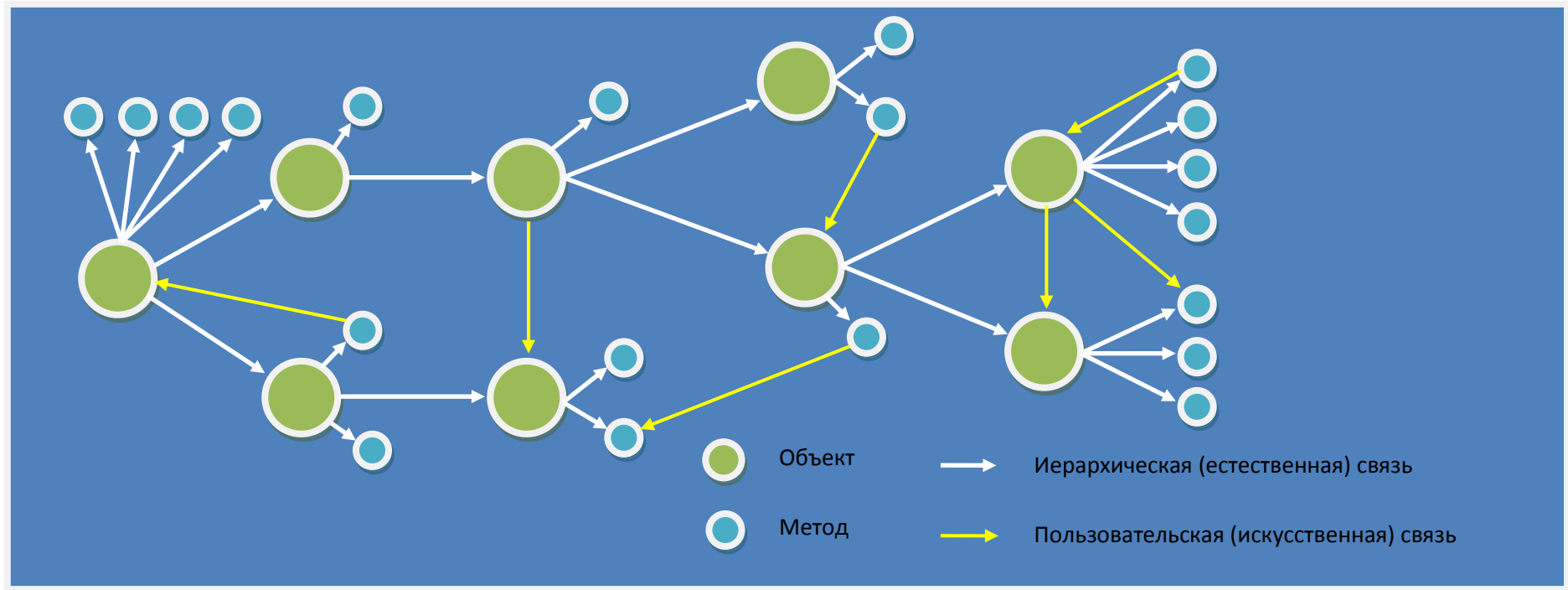
Создание понятийной базы промежуточного языка предусматривает три этапа:

1. Выделение общего понятийного поля объектов принадлежащих языкам C#, Objective C и Java и соответствующим средам разработки.
2. Создание универсального набора расширенных понятий позволяющего описывать алгоритмы и логические структуры.
3. Создание полного набора всеобъемлющих понятий, представляющего объединение всех понятийных областей используемых языков программирования при помощи набора расширенных понятий создаваемого в п.2.



Описание принципов генерации программного кода на промежуточном языке Brick:

Основной идеей в генерации исходного кода для трансляции, является то, что при использовании любого объектно-ориентированного декларативного языка, любой объект, в том числе программа целиком может быть представлен в виде графа древовидной структуры:



Программа в графическом представлении выглядит как сложный граф в котором структура подчиненности узлов является древовидной, а связи между узлами могут иметь произвольное направление, источники и адресатов.