УДК 004.415.2+004.414.38

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ МОБИЛЬНОГО ОБУЧЕНИЯ ДЛЯ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ ПРОЕКТНОГО МЕНЕДЖЕРА

М. М. Абызов[1], И. С. Шахова[2]

Высшая школа информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета

[1]mishabuzov@gmail.com, [2]is@it.kfu.ru

Аннотация

Представлен обзор способов измерения прогресса разработки программного продукта в рамках гибкой методологии SCRUM, а также дано описание разработки программного инструмента, отслеживающего текущее состояние проекта по временным характеристикам. Анализируя их, такой инструмент способен подсказать проектному менеджеру, на что именно ему следует обратить внимание в текущей проектной обстановке, и помочь в выборе действий для достижения эффективных результатов.

***Ключевые слова**: управление проектами, проектный менеджер, обучение проектных менеджеров, мобильное приложение, SCRUM, User Story, Story Point, Sprint, Sprint Backlog, Agile reporting, Mobile Learning*

ВВЕДЕНИЕ

На сегодняшний день внедрение гибких методологий разработки в процесс производства является весьма распространенной практикой. По данным исследований всемирной организации Project Management Institute, в 2017 году почти ¾ компаний так или иначе использовали гибкие методологии разработки в процессе управления своими проектами [1].

Ключевым принципом гибких методологий разработки является то, что они позволяют проводить периодическую корректировку действий, которая может привести к устойчивым и эффективным результатам [2]. В то же время, по данным статистики за 2017 год, примерно 97 млн. долларов на каждый инвестированный миллиард теряются в результате недостаточно эффективного процес-

са управления проектами [1]. Именно поэтому обучению проектных менеджеров следует уделять особое внимание.

Успех или неудача проекта во многом зависят от способности проектного менеджера правильно расставлять приоритеты между бюджетом проекта, его качеством и временем разработки, а также вовремя разрешать конфликты, возникающие в ходе жизненного цикла проекта [3]. В течение всего цикла разработки программного продукта уровень важности каждого из приоритетов может несколько раз изменяться. Всё это отражается и на временных рамках. Например, переоценка графика может оказаться фатальной для проекта, когда бюджет становится решающим фактором, и наоборот. Аналогично требования к качеству продукта должны быть изменены в определенных ситуациях, чтобы соответствовать реальности.

Чтобы правильно оценить ситуацию при довольно быстро меняющихся приоритетах, проектному менеджеру, безусловно, необходим опыт. Начинающим менеджерам может быть крайне трудно оценить текущую обстановку и принять решение, благодаря которому разработка программного продукта будет завершена точно в срок, и сам программный продукт при этом будет соответствовать основным требованиям заказчика.

КЛЮЧЕВЫЕ ОСОБЕННОСТИ МЕТОДОЛОГИИ SCRUM

Согласно исследованию, проведенному Agile Survey, наиболее распространенным методом среди гибких методологий разработки является Scrum (используемый 66% из 6042 опрошенных респондентов) [4]. В рамках того же опроса было установлено, что потеря контроля в управлении проектом является одной из самых серьезных проблем всех гибких методологий. Поэтому постоянное наблюдение за происходящими событиями в проекте имеет решающее значение для обеспечения видимости, контроля и адаптации.

Далее будут рассмотрены общие принципы и ключевые характеристики методологии Scrum. В идеале команда разработки в рамках методологии Scrum не должна превышать 9 человек – с большим числом участников, как правило, трудно поддерживать связь, что снижает эффективность данной методологии [5]. Каждый спринт может длиться не более 1 месяца и начинается с планового собрания, на котором обсуждаются задачи предстоящего спринта. Их выбирают

из общего списка задач, называемого Product Backlog. После этого участники команды задают время, необходимое для реализации проекта. В конце каждого спринта происходят обзор всех выполненных задач и демонстрация результата. Также в Scrum широко распространена практика ежедневных собраний, в ходе которых обсуждаются текущий прогресс и возможные препятствия при реализации задач.

КРИТЕРИИ ЭФФЕКТИВНОСТИ РЕЗУЛЬТАТОВ

Существуют хорошо известные критерии измерения прогресса в рамках гибкой методологии Scrum – это скорость и объем оставшейся работы [4]. Далее эти критерии будут рассмотрены на примерах соответствующих им графиков.

1) *Диаграмма скорости выполнения проекта*

Под скоростью выполнения подразумевается объем работы, который необходимо выполнить в текущем спринте [4]. Она выражается в Story Point. Как правило, 1 Story Point соответствует 1 рабочему дню. Планируемая скорость выполнения задач оценивается командой в начале каждого спринта и рассчитывается так, чтобы успеть выполнить все User Story (функциональные требования), запланированные на спринт. Фактическую же скорость можно вычислить в конце спринта путем суммирования Story Point для всех User Story, принятых владельцем продукта.

Рассмотрим, к примеру, диаграмму, отражающую скорость выполнения задач спринта на примере проекта создания веб-сайта для словенской издательской компании.

Результаты исследования проекта, отраженного на диаграмме Рис. 1, показали, что фактическая скорость выполнения спринта оказывалась всегда меньше запланированной [4]. Для первых спринтов это может быть объяснено отсутствием опыта в рамках нового проекта, поэтому и планируемая скорость вначале оценивалась, просто считая рабочий день (т. е. Story Point) равным 6 часам эффективной работы. Но самая низкая скорость оказалась в пятом спринте, когда в команду были добавлены два новых разработчика, которые по задумке должны были увеличить объем выполненной работы, но вместо этого привели проект к срыву, снизив производительность других членов команды. Существенное различие между запланированной и фактической скоростями в спринте 6 объясняется переоценкой реальных возможностей команды: вместо того, чтобы

приспособить оценку к фактическим достижениям в предыдущих спринтах, команда поддалась давлению приближающегося срока и обещала предоставить больше функциональности, чем это было реально возможно.

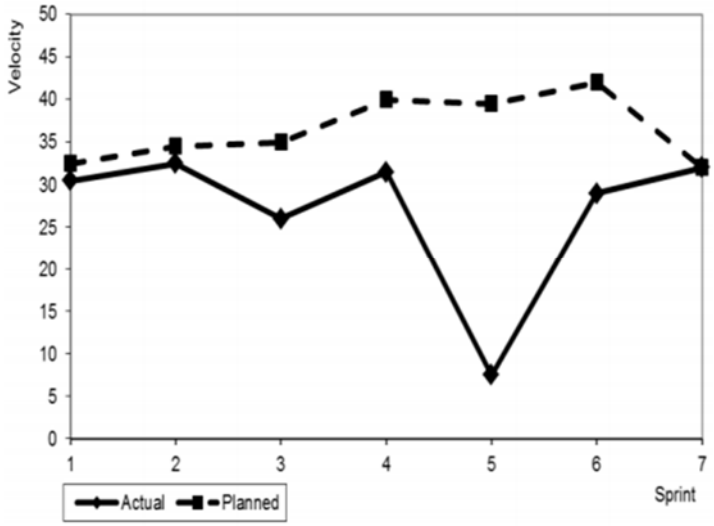

Рис. 1. Диаграмма скорости выполнения спринтов. Отражает действительную и запланированную скорости [4]

2) *Диаграмма сгорания для проекта*

Диаграмма сгорания, представленная на Рис. 2, сигнализирует о том, что у команды не получилось завершить весь проект в течение 7 спринтов, как ожидалось изначально. Основной причиной этого, как правило, являются новые требования, которые изначально не были зафиксированы в Product Backlog, но, тем не менее, постоянно добавлялись заказчиком в течение всего проекта [4]. В таком случае необходимо пересмотреть функционал проекта, чтобы выделить основные моменты и завершить проект в более приемлемые сроки. Используя такой подход, издательская компания пересмотрела содержимое Product Backlog и успешно выпустила проект после 9 спринтов. Следует отметить, что описанный подход вполне можно считать оправданным: куда лучше пересмотреть требования и выпустить продукт во вновь установленные сроки, чем рисковать не выпустить его вовсе. Тем более, что, согласно исследованиям в области разработки программного обеспечения, почти 65% функционала выпускаемых программ используются в действительности редко, а часть не используется вообще [6].

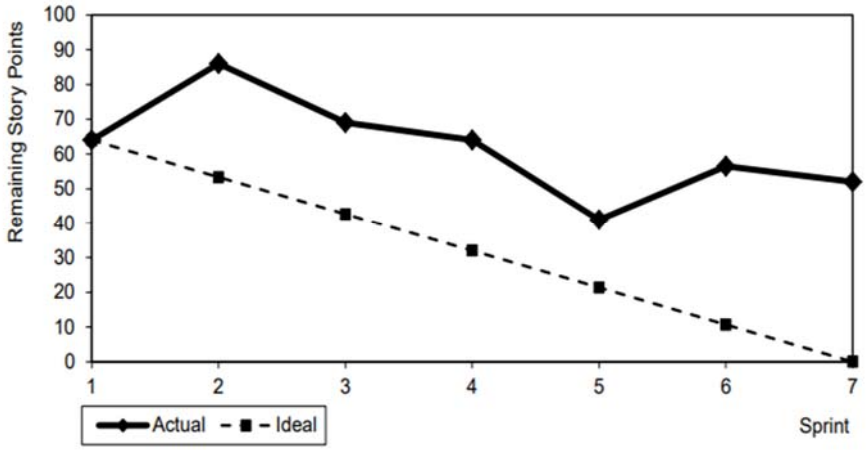

Рис. 2. Диаграмма сгорания для проекта к началу 7-го спринта [4]

3) *Диаграмма сгорания для спринта*

Эта диаграмма очень похожа на диаграмму сгорания для всего проекта, но вместо общей картины в ней отображено количество работы, которую необходимо выполнить, чтобы завершить текущий спринт. На горизонтальной линии отображается количество дней спринта, в то время как вертикальная линия показывает количество оставшихся рабочих часов. Диаграмма обновляется каждый день, суммируя оценки для всех задач. Текущая линия тренда отражает, сумеет ли команда выполнить все задачи до конца спринта. В отличие от диаграммы на Рис. 2, диаграмма сгорания для текущего спринта на Рис. 3 отражает более детально то, что команда выполнила почти все поставленные перед ней задачи в рамках текущего спринта [4].

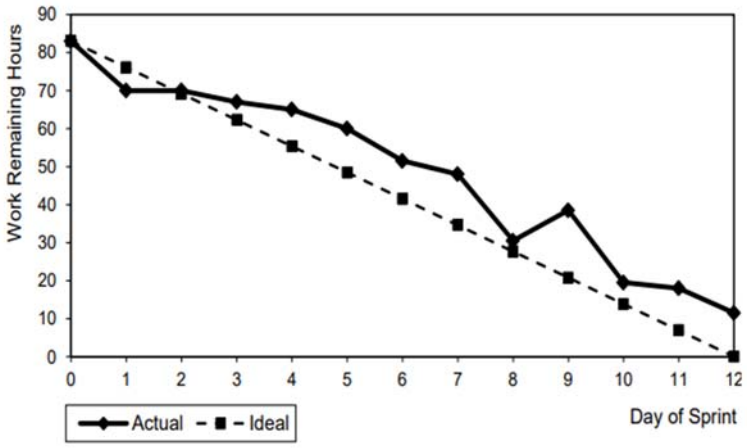

Рис. 3. Пример диаграммы сгорания для спринта [4]

На основе анализа данной диаграммы проектный менеджер может сделать и другие выводы о том, как повысить эффективность работы своей команды.

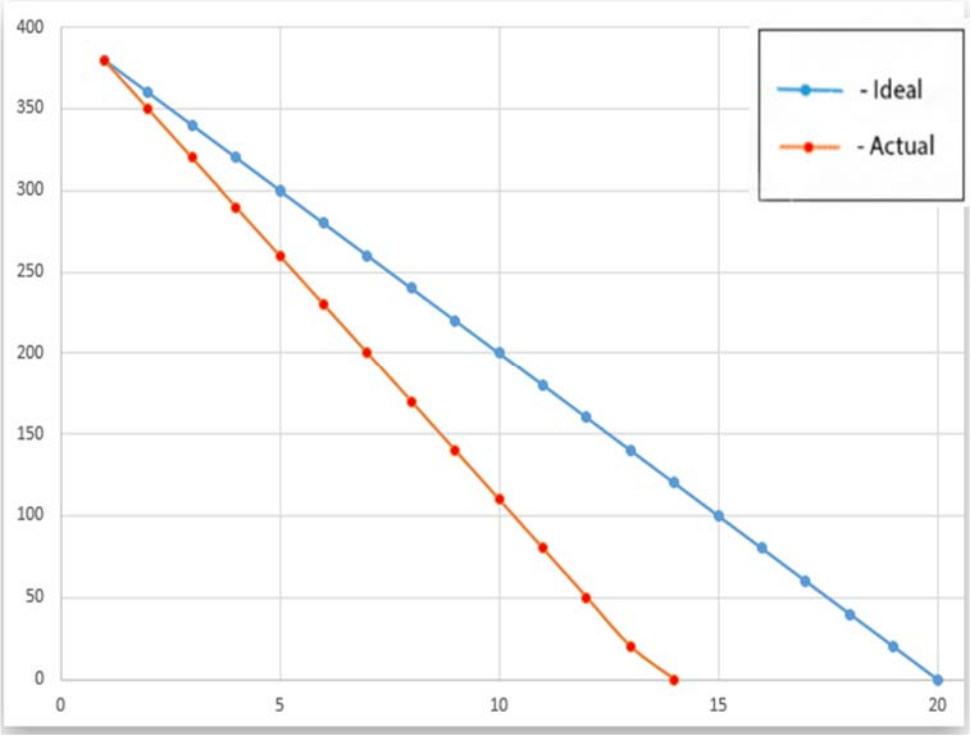

Рис. 4. Диаграмма сгорания спринта – завершен намного раньше срока [7]

Например, на диаграмме Рис. 4 отчетливо видно, что команда завершила работу по спринту раньше заявленного срока. Конечно, такой результат является позитивным для проекта в целом, однако же по нему можно судить и о ряде имеющихся проблем в команде [7]:

- оценка предстоящей работы была выполнена неправильно, команда сильно перестраховалась;
- в ходе выполнения спринта не были добавлены новые задачи; как правило, именно это и рекомендуется сделать проектному менеджеру в ситуации, изображенной на Рис. 4.

Ещё одним, более негативным вариантом развития событий является ситуация, когда команда сильно не успевает выполнять текущие задачи. Тогда диаграмма сжигания задач спринта может выглядеть примерно так, как показано на Рис. 5.

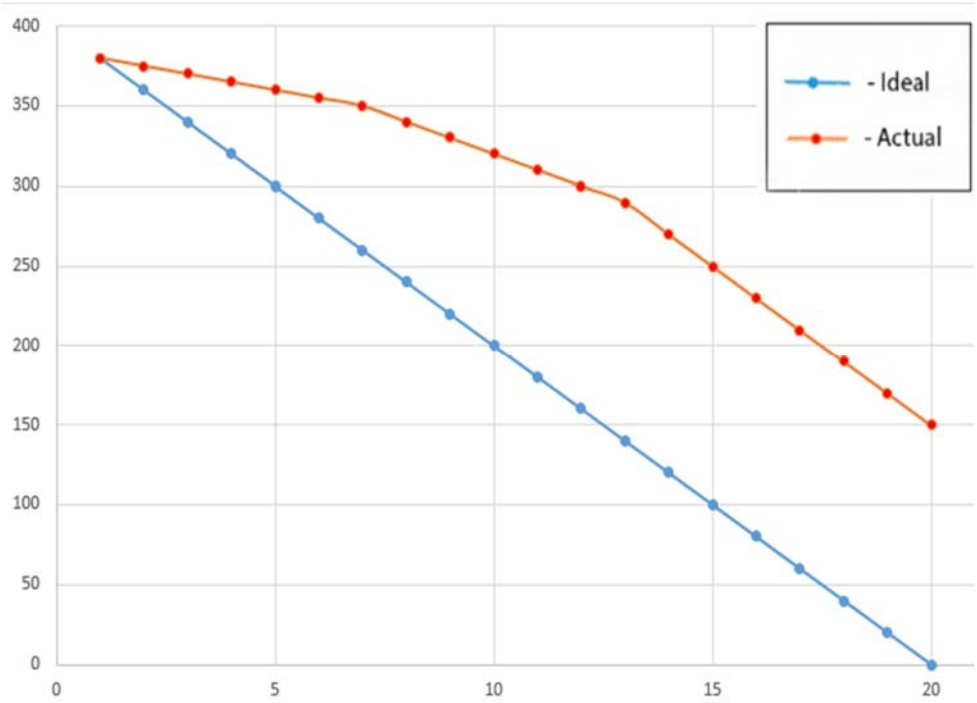

Рис. 5. Диаграмма сгорания спринта – сильное отставание в графике выполнения задач [7]

Данная диаграмма свидетельствует о недостаточной эффективности ежедневных собраний команды в рамках методологии Scrum. Возможно, здесь имело место постоянное добавление задач во время спринта, либо же часть задач была выполнена не полностью [7]. В любом случае менеджеру проекта следует уделить особое внимание ситуации, о которой свидетельствует график на Рис. 5.

Отметим, что любое отклонение вверх графика реального выполнения задач от запланированного графика выполнения задач спринта, согласно одному из постулатов методологии Scrum [5], уже является проблемой: её обязательно нужно обсуждать и решать в рамках проектных встреч.

4) *Диаграмма совокупного потока*

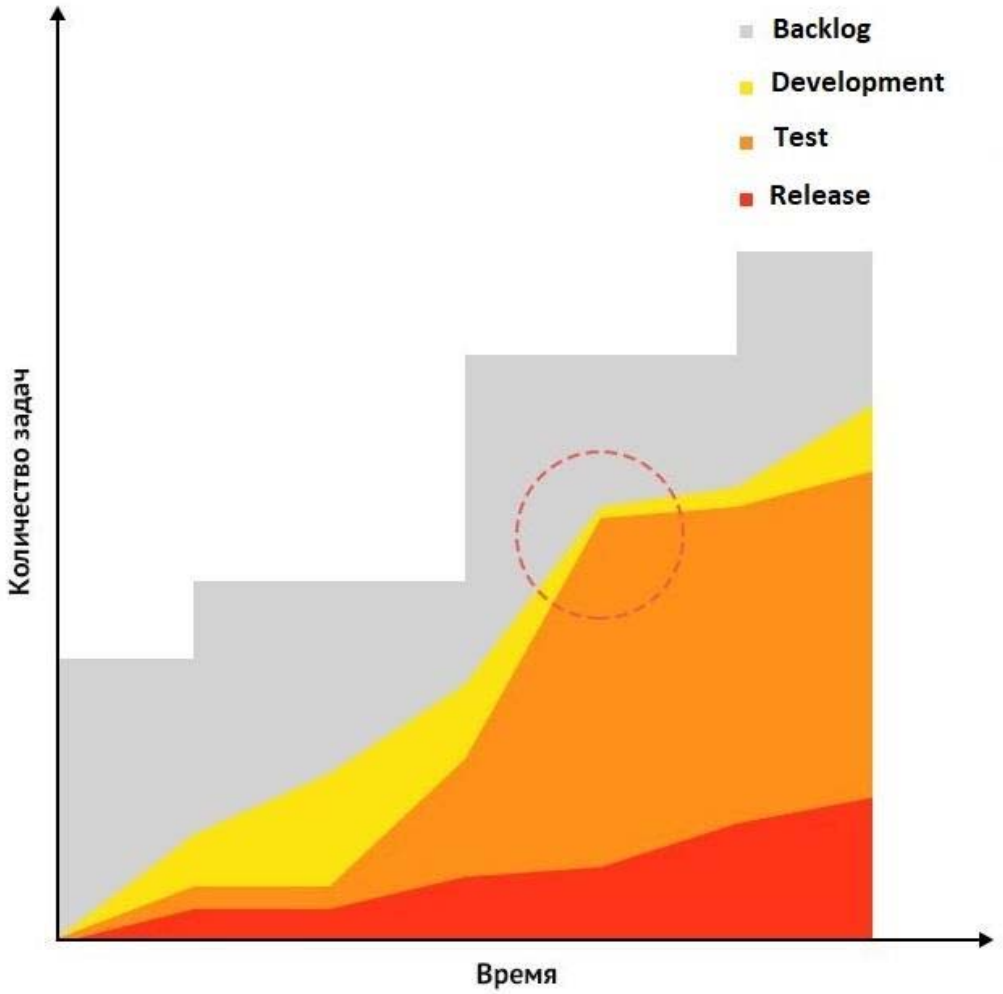

Рис. 6. Диаграмма совокупного потока [8]

На Рис. 6 приведен пример диаграммы совокупного потока. Она представляет собой график, визуализирующий количество задач с тем или иным статусом во времени [8]. Линии, иллюстрирующие разные статусы, либо приближаются друг к другу, либо отдаляются. Если одна линия начинает почти полностью перекрывать другую, то это говорит о том, что в работе над проектом образуются простои. Например, на данном графике изображена ситуация, когда разработчики не успевают обеспечить задачами отдел тестирования. В этом случае менеджеру необходимо повысить производительность команды разработки либо сократить количество новых задач – по ситуации.

ИСПОЛЬЗОВАНИЕ ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ В СИСТЕМАХ УПРАВЛЕНИЯ ПРОЕКТАМИ

Для работы над проектом менеджеры, как правило, используют различные системы управления с многопользовательским доступом для распределения текущих задач и отслеживания статуса их выполнения [9]. Такие системы представляют собой пользовательский интерфейс с различным набором инструментов, в том числе и широко применяемых в гибких методологиях разработки, как, например, Scrum- и Kanban-доски.

Как уже было показано выше, имея статистику задач по проекту, данные по срокам их исполнения, связанности задач друг с другом, данные об исполнителях и т. д., можно отразить приблизительную динамику выполнения проекта, позволяющую на основе предыдущих показателей предсказать, завершится ли текущий спринт в установленные для него сроки или часть задач останется незавершенной.

Некоторые системы, как, например, Atlassian Jira, предлагают такой функционал автоматического построения графиков [10]. Однако же начинающий проектный менеджер далеко не всегда способен понять, на что именно нужно обратить внимание на данный момент, исходя из имеющихся показателей. Поэтому было решено разработать программный инструмент, который сообщал бы начинающим менеджерам проектов о наиболее важных отклонениях, возникающих в ходе разработки программного продукта.

Принцип работы и архитектура приложения

Созданное приложение имеет клиент-серверную архитектуру, и все новые данные по мере их создания сразу же отправляются на сервер, где хранятся в удаленной базе данных. Между клиентом и сервером ведется синхронизация, поэтому если произошло какое-либо значимое для проекта событие, то клиент сразу же будет оповещен.

Само приложение представляет собой мобильный клиент для системы управления проектами. В нем реализованы многие стандартные функции, необходимые проектному менеджеру: создание новых задач, спринтов, проектов, их редактирование, мониторинг активности, модуль документации и некоторые другие. Кроме всего упомянутого, приложение имеет модуль статистики, показывающий графики с текущим состоянием проекта, а также модуль оповещений,

целью которого является уведомить проектного менеджера, когда состояние проекта начинает отклоняться от запланированного, и посоветовать ему предпринять какие-либо меры в зависимости от сложившейся ситуации.

Виды оповещений

Выше был представлен способ анализа состояния выполнения проекта на основе нескольких диаграмм. По данным для их построения и о сроках выполнения задач можно выделить ключевые моменты в отклонении промежуточных результатов проекта от изначально запланированных. Рассмотрим некоторые из них:

1) ***Недооценка времени выполнения критических задач***

Задачи с высоким для проекта приоритетом должны выполняться в первую очередь, т. к. именно от их реализации напрямую зависит главный функционал проекта. Поэтому особо важно следить за статусом выполнения таких задач и как можно быстрее реагировать на возникающие препятствия в их реализации.

Таким образом, если задача с высоким приоритетом, запланированная на текущий спринт, будет по каким-либо причинам не выполнена в запланированные сроки, приложение уведомит об этом проектного менеджера и посоветует ему рассмотреть эту проблему на ежедневной встрече. Механизм работы уведомления для этого случая представлен на блок-схеме Рис. 7, где **n** – количество задач в спринте, **taskList** – коллекция задач спринта.

2) ***Отставание команды от графика выполнения текущего спринта***

Распределение задач в команде может быть неравномерным – играют роль многие факторы, например, опыт члена команды. Одному разработчику можно делегировать сразу несколько задач на текущий спринт, а затем добавлять новые или, наоборот, передать задачу для выполнения от одного разработчика другому. Вполне возможна ситуация, что у одного члена команды ближе к концу спринта окажется задач больше, чем он рассчитывает выполнить. Тогда часть его задач нужно передать другим разработчикам. Если такая ситуация возникает сразу у всей команды разработки, то это свидетельствует о явной недооценке ситуации по спринту в целом.

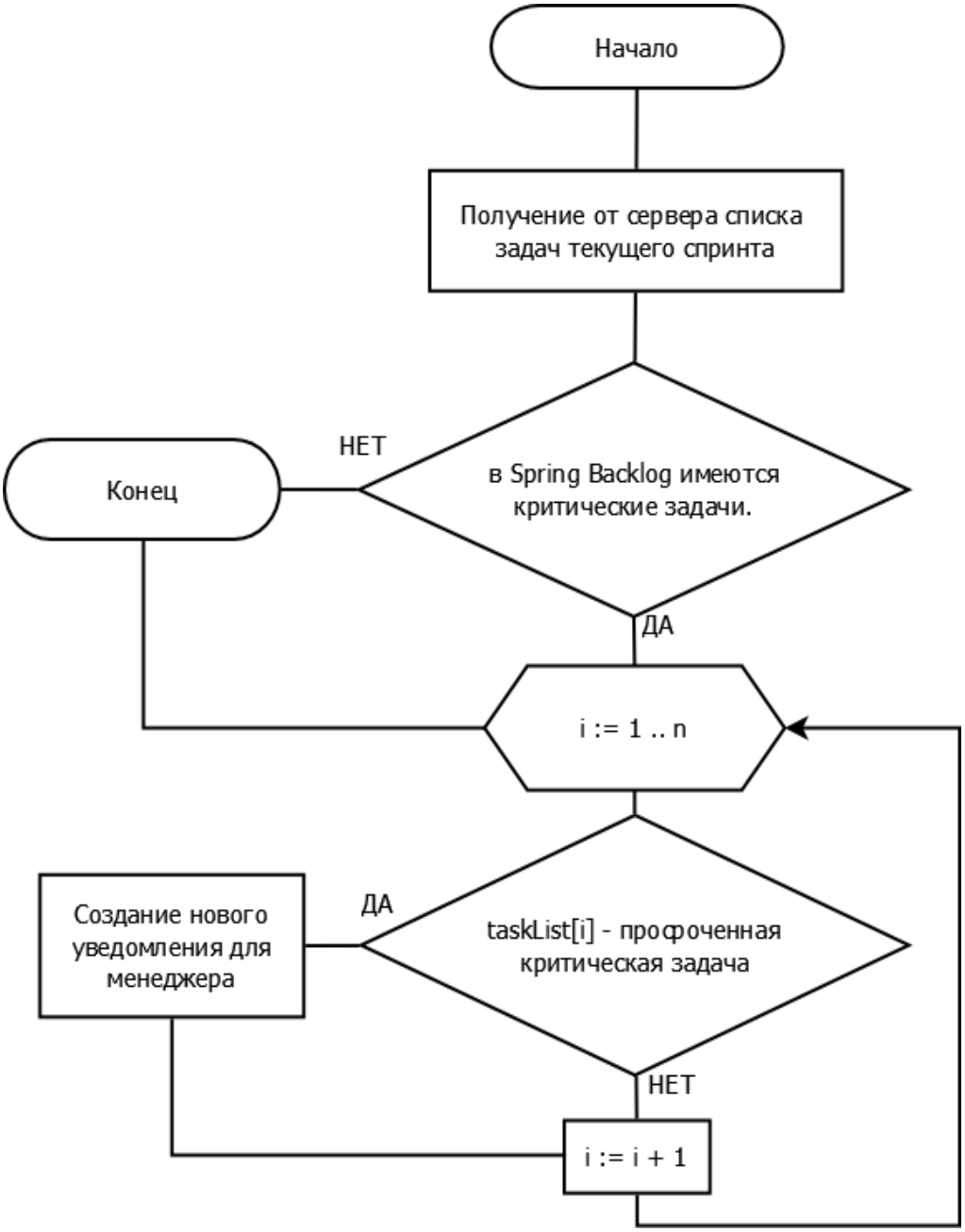

Рис. 7. Блок-схема создания уведомлений о невыполненных критических задачах

Таким образом, приложение уведомит менеджера в следующих случаях:

- если у члена команды сумма Story Point всех невыполненных задач окажется больше, чем осталось дней до завершения спринта;
- если больше, чем у 50% членов команды (в этом случае перераспределение всех задач между участниками уже маловероятно) сумма Story Point всех невыполненных задач окажется больше, чем осталось дней до завершения

спринта, приложение уведомит проектного менеджера о необходимости предпринять меры для ускорения процесса разработки, например, пересмотреть приоритет задач, чтобы успеть выполнить наиболее важные, увеличить ресурсы или производительность работы текущих исполнителей.

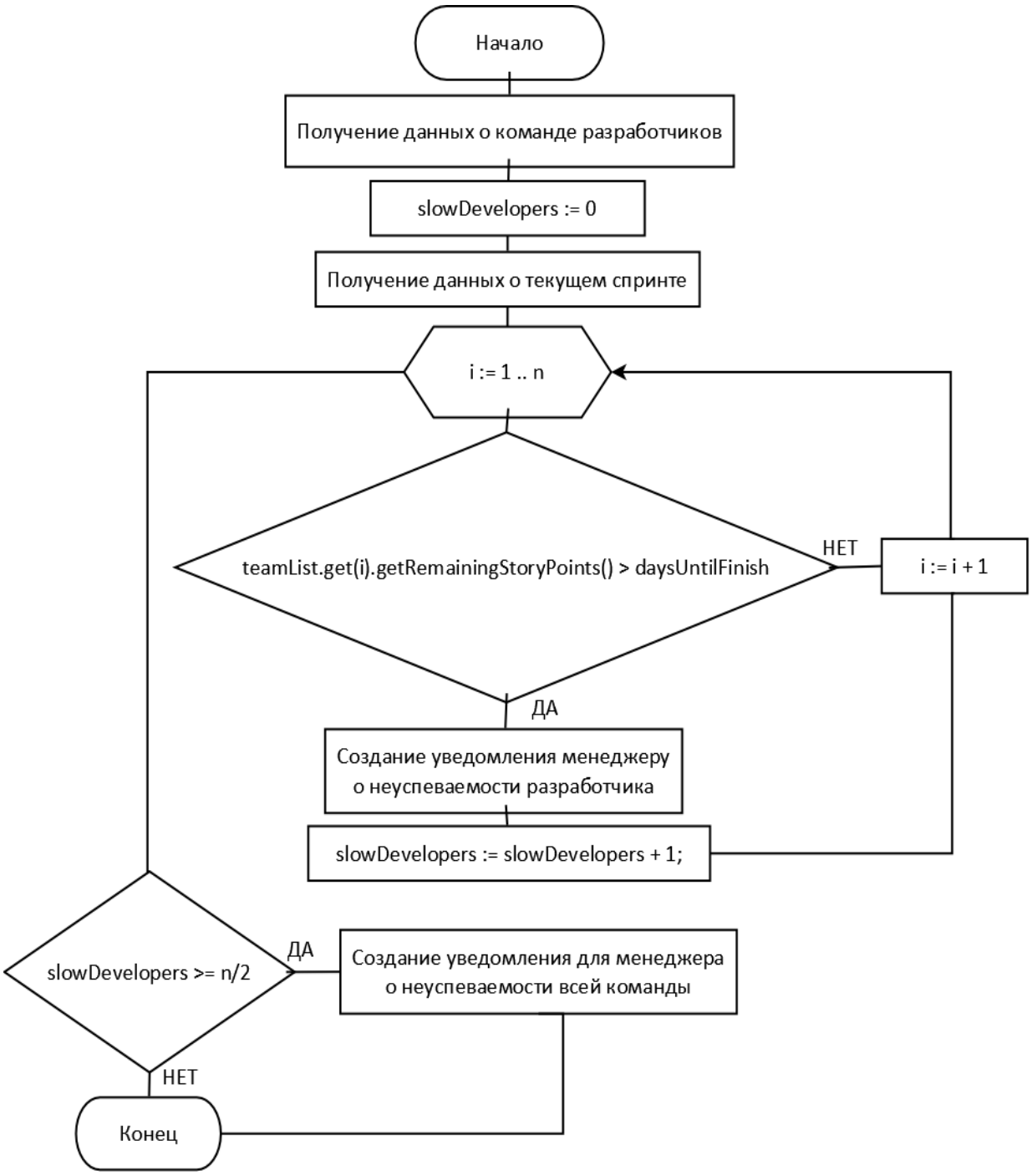

Рис. 8. Блок-схема создания уведомлений об отставании разработчиков от запланированного графика

Механизм создания уведомления для случаев отставания разработчиков от запланированного графика представлен на блок-схеме Рис. 8, где **n** – число разработчиков в команде, **teamList** – коллекция с данными о разработчиках, **getRemainingStoryPoints()** – сумма Story point всех оставшихся задач у конкретного разработчика, **slowDevelopers** – количество неуспевающих разработчиков, **daysUntilFinish** – число дней до конца текущего спринта.

3) ***Переоценка текущего спринта***

Если 70% всех задач, включая критические, будет выполнено к середине спринта или же все баги и User Story будут закрыты ещё до конца спринта, то это может свидетельствовать о переоценке командой поставленных задач. В этом случае приложение напомнит менеджеру о необходимости на ежедневной встрече расспросить о возможности добавления новых задач в Sprint Backlog. Блок-схема Рис. 9 демонстрирует принцип создания уведомлений для менеджера о возможности добавления новых задач.

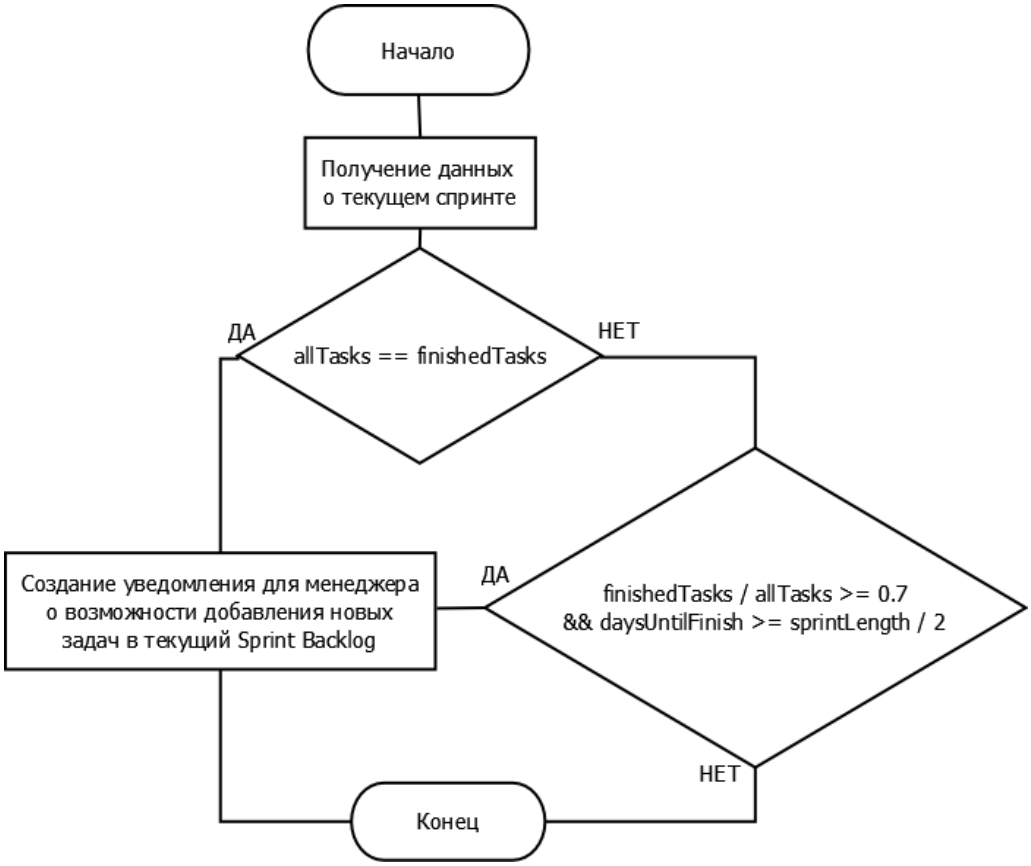

Рис. 9. Блок-схема создания уведомлений о возможности пополнения текущего Sprint Backlog

На данном рисунке **sprintLength** – количество дней в текущем спринте, **daysUntilFinish** – число дней до конца спринта, **allTasks** – число задач в текущем Sprint Backlog, **finishedTasks** – число выполненных задач текущего Sprint Backlog.

4) ***Изменение состава команды разработки***

Изменения в команде разработки во многих случаях могут сопровождаться резким падением производительности всей команды в целом, причем неважно, будут ли это уход члена команды или добавление новых участников. Один из таких случаев, когда добавление новых разработчиков снизило производительность команды в целом, был рассмотрен в статье ранее. К новичкам в проекте необходим индивидуальный подход. Перед менеджером в данном случае стоит задача посвятить их в особенности проекта, при этом сделав так, чтобы их появление не привело к сильному падению производительности работы всей команды.

Рис. 10. Блок-схема создания уведомлений для контроля новых членов команды разработки

- При появлении новых членов команды приложение покажет менеджеру уведомление о необходимости приставить к ним наставника, разбирающегося в проекте, а также перед каждым ежедневным митингом на протяжении первого спринта с новыми участниками у менеджера будет появляться уведомление о необходимости расспросить новичков о текущем прогрессе в понимании проекта и его технологий. Механизм создания уведомлений для данного случая представлен на блок-схеме Рис. 10, где **sprintsFinished** – количество завершенных спринтов.
- Если же участников в проекте становится меньше 3 или больше 9 (рекомендуемые количества членов команды разработки в рамках методологии Scrum [5]), то проектный менеджер получит уведомление о недостатке людей в команде (вследствие чего есть риск не успеть выполнить все задачи) либо о слишком большом числе разработчиков (вследствие чего могут возникнуть проблемы с коммуникацией между ними). Блок-схема для этой ситуации представлена на Рис. 11, где **developersCount** – количество разработчиков в команде.

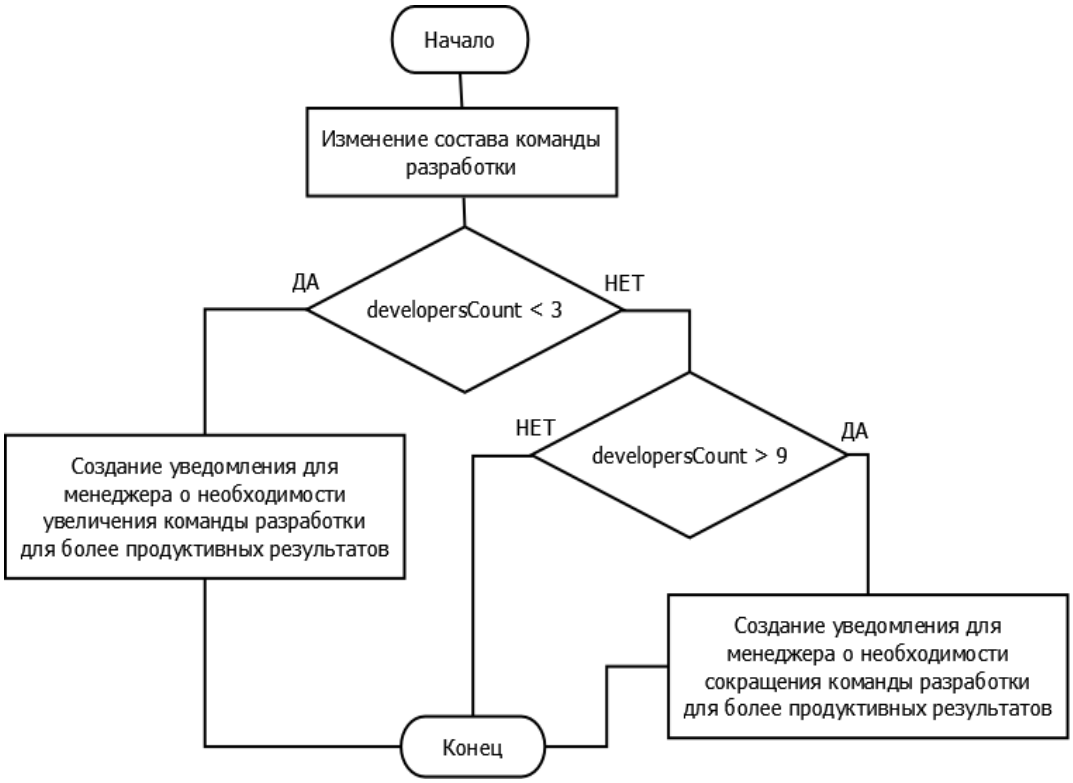

Рис. 11. Блок-схема, иллюстрирующая процесс создания уведомлений о неоптимальном количестве членов команды разработки

5) ***Простои в команде разработки***

При рассмотрении диаграммы совокупного потока в предыдущем разделе была отмечена ситуация простоя отдела разработки. В приложении это отслеживается путем сравнивания количества User Story и багов у разработчиков и тестировщиков за весь спринт (либо за весь процесс разработки продукта).

Таким образом, если по ходу спринта количество задач у подразделений начинает совпадать либо отличаться на незначительное число (в приложении за такое число было взято количество человек в команде разработки), то менеджер получит уведомление о простое в команде разработчиков и о том, что необходимо повысить их производительность.

Блок-схема создания уведомлений о простое разработчиков представлена на Рис. 12, где **developersCount** – количество разработчиков в команде, **developersTasks** – общее количество решаемых задач, **testTasks** – общее количество задач для тестировщиков.

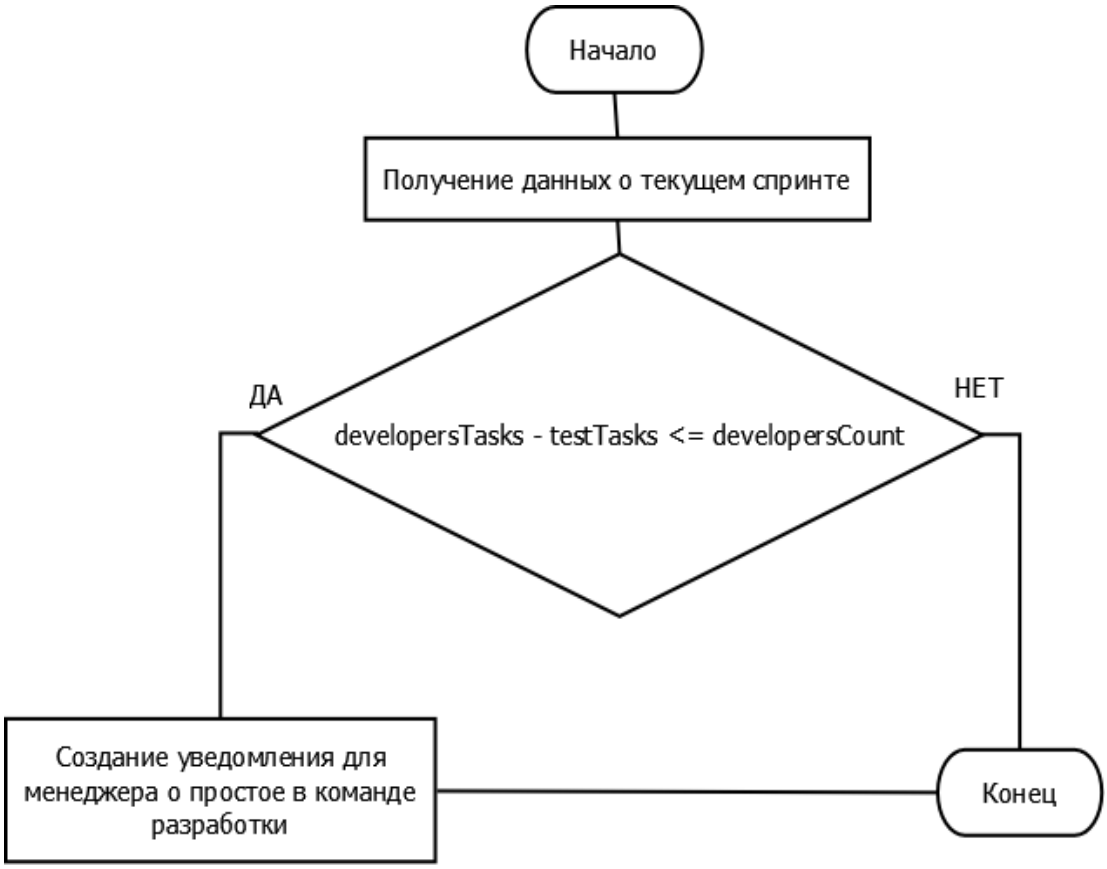

Рис. 12. Блок-схема создания уведомлений о простое команды разработчиков

6) ***Переполнение Product Backlog***

В случае изменения заказчиком требований в Product Backlog могут быть добавлены новые задачи либо же могут быть изменены требования к уже существующим. Такая ситуация рассматривалась в предыдущем разделе и оказалась весьма критичной для проекта: потребовалось ещё 2 незапланированных спринта для завершения разработки.

Таким образом, в этих случаях приложение сразу будет уведомлять менеджера:

- о необходимости изучить новые требования и задачи, как только они будут добавлены;
- если добавление новых требований повлекло за собой существенное увеличение Product Backlog более чем на 20%, то менеджеру в такой ситуации будет предложено сделать полную переоценку первоочередного функционала, увеличить ресурсы либо повысить производительность работы текущих исполнителей.

На Рис. 13 изображена блок-схема создания уведомлений в приложении о добавлении новых задач в Product Backlog, где **beginBacklogCount –** количество задач в Product Backlog к началу очередного спринта, **currentBacklogCount –** количество задач в Product Backlog на момент синхронизации приложения с удаленным сервером.

ЗАКЛЮЧЕНИЕ

В статье проанализированы лишь некоторые данные, на основании которых менеджер проектов может судить об эффективности процесса разработки программного продукта. Приведены примеры визуализации этих данных на графиках, а также разобран механизм уведомления в мобильном приложении, работающий на основе анализа рассмотренных проектных данных. Такое приложение в будущем при активном его внедрении и расширении способно помочь начинающим менеджерам в процессе их обучения.

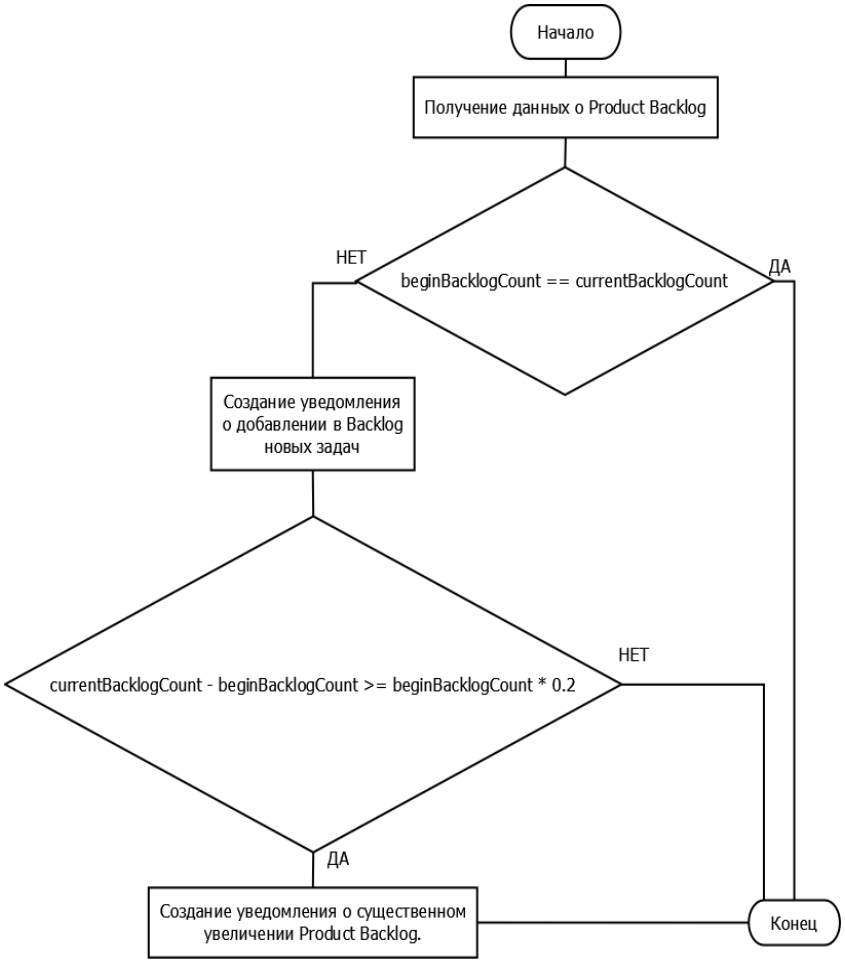

Рис. 13. Блок-схема создания уведомлений о добавлении в Product Backlog новых задач

СПИСОК ЛИТЕРАТУРЫ

1. Success Rates Rise. Transforming the high cost of low performance // Обзор 9-го глобального исследования в области управления проектами. Project Management Institute, 2017. URL: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

2. *Clinton J. Woodward, Andrew Cain, Shannon Pace, Allan Jones and Joost Funke Kupper.* Helping Students Track Learning Progress Using Burn Down Charts // 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), 26–29 August 2013, Bali Dynasty Resort, Kuta, Indonesia.

3. *Paul O. Gaddis.* The Project Manager // Harvard Business Review (May–June 1959).

4. *Mahnic V., Zabkar N.* Measuring Progress of Scrum-based Software Projects // Elektronika i elektrotechnika, ISSN 1392-1215. 2012. V. 18, No 8.

5. *Ken Schwaber and Jeff Sutherland.* The Scrum Guide™ // The Definitive Guide to Scrum. URL: http://www.scrumguides.org/ docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100

6. *Victor Szalvay.* An Introduction to Agile Software Development // Copyright © 2004 Danube Technologies, Inc. URL: http://www.danube.com/docs/Intro_to_ Agile.pdf

7. Система управления проектами “Scrum time”. URL: https://ru.scrum-time.com/infobase/burndown-chart.php

8. Scrum – студия “Сибирикс”. URL: https://blog.sibirix.ru/2014/11/27/ graphs/

9. *Alok Mishra, Deepti Mishra.* Software Project Management Tools: A Brief Comparative View // ACM SIGSOFT Software Engineering Notes, 2013. V. 38, No 3, May.

10. Система управления проектами “Jira Software”. URL: https://ru.atlassian. com/software/jira/features.org

PROJECT MANAGER’S COMPETENCY BUILDING USING MOBILE LEARNING TECHNOLOGIES

M. M. Abyzov[1], I. S. Shakhova[2]

Higher School of Information Technologies and Intelligent Systems at Kazan Federal University

[1]mishabuzov@gmail.com, [2]is@it.kfu.ru

Abstract

An overview of the methods for measuring progress of software development within flexible SCRUM methodology is given, as well as a description of a software tool development that monitors current status of project by time characteristics. The tool based on the characteristics can notificate project manager about events he have to pay attention in current project situation and help him to achieve effective results.

Keywords: *project management, project manager, training of project managers, mobile application, SCRUM, User Story, Story Point, Sprint, Sprint Backlog, Agile reporting, Mobile Learning.*

REFERENCES

1. Success Rates Rise. Transforming the high cost of low performance // Обзор 9-го глобального исследования в области управления проектами. Project Management Institute, 2017. URL: https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf

2. *Clinton J. Woodward, Andrew Cain, Shannon Pace, Allan Jones and Joost Funke Kupper.* Helping Students Track Learning Progress Using Burn Down Charts // 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE), 26–29 August 2013, Bali Dynasty Resort, Kuta, Indonesia.

3. *Paul O. Gaddis.* The Project Manager // Harvard Business Review (May–June 1959).

4. *Mahnic V., Zabkar N.* Measuring Progress of Scrum-based Software Projects // Elektronika i elektrotechnika, ISSN 1392-1215. 2012. V. 18, No 8.

5. *Ken Schwaber and Jeff Sutherland.* The Scrum Guide™ // The Definitive Guide to Scrum. URL: http://www.scrumguides.org/ docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100

6. *Victor Szalvay.* An Introduction to Agile Software Development // Copyright © 2004 Danube Technologies, Inc. URL: http://www.danube.com/docs/Intro_to_Agile.pdf

7. Project management system “Scrum time”. URL: https://ru.scrum-time.com/infobase/burndown-chart.php

8. Scrum – studiya “Sibiriks”. URL: https://blog.sibirix.ru/2014/11/27/ graphs/

9. *Alok Mishra, Deepti Mishra.* Software Project Management Tools: A Brief Comparative View // ACM SIGSOFT Software Engineering Notes, 2013. V. 38, No 3, May.

10. Issue tracking system “Jira Software”. URL: https://ru.atlassian.com /software/jira/features.org

СВЕДЕНИЯ ОБ АВТОРАХ

АБЫЗОВ Михаил Михайлович – студент Высшей школы информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, андроид-разработчик.

Mikhail Mikhailovich ABYZOV – student of the Higher School of Information Technologies and Intelligent Systems at Kazan Federal University, Android developer.

email: mishabuzov@gmail.com

ШАХОВА Ирина Сергеевна – ассистент кафедры программной инженерии Высшей школы информационных технологий и интеллектуальных систем Казанского федерального университета. Сфера научных интересов – цифровые образовательные системы, индивидуализация образования, мобильное обучение.

Irina Sergeevna SHAKHOVA – teacher of the Higher School of Information Technologies and Intelligent Systems at Kazan Federal University. Research interests include digital educational systems, individualization of education, mobile learning.

email: is@it.kfu.ru

Материал поступил в редакцию 31 мая 2018 года