

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ОПТИМИЗАЦИИ СОСТАВА КОМАНДЫ ИТ-ПРОЕКТА

Васильева Л. В., Шелест А. И.

Исследованы возможности управления трудовыми ресурсами с учетом их квалификации при их распределении на проекты (задачи). Выявлены факторы, влияющие на критерии оптимальности. Построены математические модели для оптимизации состава команды ИТ-проекта для двух типов задач: critical и minor. Приведена диаграмма прецедентов бизнес-процесса программно-методического комплекса для управления ИТ-проектами. Проведенные предварительные расчеты показали эффективность предложенного подхода, при использовании которого может быть получена возможность минимизации затрат для задач типа minor и уменьшение времени на процесс разработки задач типа critical.

Досліджено можливості управління трудовими ресурсами з урахуванням їх кваліфікації при їх розподілі на проекти (завдання). Виявлено фактори, що впливають на критерії оптимальності. Побудовано математичні моделі для оптимізації складу команди ІТ-проекту для двох типів завдань: critical і minor. Наведено діаграму прецедентів бізнес-процесу програмно-методичного комплексу для управління ІТ-проектами. Проведені попередні розрахунки показали ефективність запропонованого підходу, при використанні якого може бути отримана можливість мінімізації витрат для завдань типу minor і зменшення часу на процес розробки завдань типу critical.

The article explores the possibilities of managing human resources taking into account their qualifications when they are allocated to projects (tasks). Factors influencing the optimality criteria are revealed. Mathematical models have been constructed to optimize the composition of the IT project team for two types of tasks: critical and minor. The diagram of precedents of business process of the program-methodical complex for management IT-projects is given. Preliminary calculations have shown the effectiveness of the proposed approach, with the use of which it can be possible to minimize costs for tasks of the minor type and reduce the time for the process of developing critical tasks.

Васильева Л. В.

канд. техн. наук, доц. кафедры КИТ ДГМА
vasileva.dgma@gmail.com

Шелест А. И.

магистр каф. КИТ ДГМА
alenashelest8@gmail.com

УДК 004.4 : 519.85

Васильева Л. В., Шелест А. И.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ОПТИМИЗАЦИИ СОСТАВА КОМАНДЫ ИТ-ПРОЕКТА

Программы и проекты имеют большую важность для многих коммерческих и государственных организаций; благодаря проектам предприятия могут увеличить прибыль, улучшить процесс создания концепции продукта, его разработки и внедрения на рынок, способствовать созданию новых или усовершенствованных средств производства, информационных систем. Проекты представляют собой инструмент развития и совершенствования организаций на всех уровнях: города, района, страны. С помощью программ и проектов образовательные и другие учреждения вводят новые и совершенствуют уже предоставляемые ими услуги.

Рабочий процесс работы над проектом любой ИТ-компании среднего бизнеса существенно отличается от работы крупной промышленной компании. Управление проектом в такой компании осуществляется одним или несколькими проектными менеджерами. Каждый заказ оформляется как новый проект, который имеет четко выраженные этапы, сроки, результаты, поэтапную заработную плату сотрудников. Проекты управляются и контролируются менеджерами. Ошибки, допущенные в одном проекте, могут стать причиной потери прибыли множества других связанных с ним проектов. Важно, чтобы каждая организация, которая несет ответственность за проект, умела эффективно им управлять.

Возможность получения значительного эффекта от реализации ИТ-проекта обуславливает важность роли предварительной оценки при выборе проекта [1].

Процессы управления проектами поддаются стандартизации и документы, которые формализуют эти процессы, получили название методологий управления проектами. Существует специальное программное обеспечение для управления проектами. Например, Wrike [2] – это система управления проектами, в том числе для создания задач, сбора команды, подключения к сервисам совместной работы, интеграции почты; обеспечивает онлайн-среду для рабочего взаимодействия, как в локальной, так и в распределенной команде, позволяет планировать проекты, определять приоритеты задач и отслеживать график их выполнения; помимо веб-сервиса, Wrike также доступен в виде приложений для iOS и Android. JIRA [3] – это система, которая подходит для отслеживания ошибок и управления проектом в компании любого размера. JIRA помогает команде обмениваться информацией и вовлекать разных сотрудников в проекты и задачи; отслеживать и фиксировать ошибки пользователей в работе с программными продуктами; обеспечивать соблюдение работы точно в срок и в рамках регламента рабочего процесса, проверять и планировать эффективность работников и назначать определённые задачи; работать вместе с коллегами с помощью инструментов совместного редактирования файлов, а также отслеживать прогресс и обновление каждой задачи команды.

Целью данной работы является исследование повышения производительности труда и снижение рисков за счет планирования и учета выполнения работ в автоматизированной системе управления, создание математической модели для оптимизации состава команды ИТ-проекта.

В настоящее время существует множество методов расчета затрат на создание IT-проекта [4], учитывающие различного вида ресурсы: технические, управленческие, трудовые. Для межпроектного перемещения работника на IT-предприятии могут быть созданы системы поддержки принятия решений [5].

В данной работе рассматривался вопрос построения оптимизационной математической модели для управления трудовыми ресурсами с учетом их квалификации. Для достижения желаемых результатов в установленные сроки и в пределах определенных денежных затрат или других важных ресурсов, необходимо качественно управлять командной разработкой проектами. Правильно сформированная команда разработчиков дает возможность наиболее эффективной работы.

Постановка задачи. Имеется некоторое количество работников – разработчиков и тестировщиков (QA-engineer), которых следует распределить по позициям в будущих проектах. Работа считается неоднородной, ее количество измеряется в часах. Каждая задача имеет сроки выполнения. Задачи подразделяются на 2 типа: *critical* и *minor*. Необходимо построить математическую модель для оптимизации состава команды на основе истории работ каждого участника системы, принимая во внимание различные приоритеты задач. Рассматривались два критерия оптимальности: время выполнения задачи и стоимость (затраты на заработную плату).

Рассмотрим задачи типа *critical*. Введем обозначения:

t_{di} – время работы разработчика d ($d = 1..m$) над задачами/подзадачами i ($i = 1..k$) указанного типа;

$T_{est(i)}$ – оценочное время выполнения задачи/ подзадачи;

$x_{di} = \sum_{i=1}^k T_{est(i)} / \sum_{i=1}^m t_{di}$ – коэффициент эффективности каждого из разработчиков;

t_{qj} – время работы QA-engineer q ($q = 1..m$) над задачами/подзадачами j ($j = 1..k$) указанного типа;

$x_{qi} = \sum_{j=1}^k T_{est(j)} / \sum_{j=1}^m t_{qj}$ – коэффициент эффективности каждого из QA-engineer

Величины t_{di} , t_{qj} , $T_{est(i)}$ относятся к ранее выполненным задачам типа *critical*.

Для задач типа *critical* наиболее важно время выполнения задачи, а стоимость можно перевести в ограничения, поэтому целевой функцией будет функция:

$$f_1 = T_{est(d)}/x_{di} + T_{est(q)}/x_{qj} \rightarrow \min.$$

Получим математическую модель задачи:

$$\begin{cases} f_1 = T_{est(d)i}/x_{di} + T_{est(q)j}/x_{qj} \rightarrow \min, \\ \left\{ \begin{array}{l} T_{est(d)i}/x_{di} + T_{est(q)j}/x_{qj} \leq T \\ x_{di} > 0 \\ y_{qj} > 0 \end{array} \right. \end{cases}, \quad (1)$$

где T – общее время (заданное) на решение новой задачи;

$T_{est(d)}$ – время, выделенное на разработку;

$T_{est(q)}$ – время, выделенное на тестирование.

Указанные выше коэффициенты эффективности каждого из работников рассчитываются на основании истории ранее выполненных задач, для чего должны быть составлены таблицы (табл. 1, 2).

Таблица 1

Время выполнения задач типа *critical*

| | Задача 1 | Задача 2 | | Задача i | | Задача k |
|----------------------------|---------------|---------------|--|---------------|--|---------------|
| Разработчик 1 | t_{11} | t_{12} | | t_{1i} | | t_{1k} |
| | | | | | | |
| Разработчик d | t_{d1} | t_{d2} | | t_{di} | | t_{dk} |
| | | | | | | |
| Разработчик m | t_{m1} | t_{m2} | | t_{mi} | | t_{mk} |
| Estimate ($T_{est(d)i}$) | $T_{est(d)1}$ | $T_{est(d)2}$ | | $T_{est(d)i}$ | | $T_{est(d)k}$ |

Таблица 2

Время проверки задач типа *critical*

| | Задача 1 | Задача 2 | | Задача j | | Задача k |
|----------------------------|---------------|---------------|--|---------------|--|---------------|
| QA engineer 1 | t_{11} | t_{12} | | t_{1j} | | t_{1k} |
| | | | | | | |
| QA engineer q | t_{q1} | t_{q2} | | t_{qj} | | t_{qk} |
| | | | | | | |
| QA engineer m | t_{m1} | t_{m2} | | t_{mj} | | t_{mk} |
| Estimate ($T_{est(q)j}$) | $T_{est(q)1}$ | $T_{est(q)2}$ | | $T_{est(q)j}$ | | $T_{est(q)k}$ |

Рассмотрим задачи типа *minor*. Введем обозначения:

S_{di} – почасовая заработная плата для разработчика d ($d = 1..m$) при работе над задачами/подзадачами i ($i = 1..k$) указанного типа;

S_{qj} – почасовая заработная плата для QA engineer q ($q = 1..m$) при работе над задачами/подзадачами j ($j = 1..k$) указанного типа;

$x_{di} = \sum_{i=1}^k T_{est} / \sum_{i=1}^m t_{di} \cdot S_{di}$ – коэффициент эффективности каждого из разработчиков;

$S_{di} = S_{avdi} / t_{di}$ – коэффициент заработной платы разработчика;

S_{avdi} – средняя (average) заработная плата разработчика;

$x_{qj} = \sum_{j=1}^k T_{est} / \sum_{j=1}^m t_{qj} \cdot S_{qj}$ – коэффициент эффективности каждого из QA engineer;

S_{avdj} – средняя (average) заработная плата QA engineer.

Для задач типа *minor* наиболее важна экономическая составляющая. Поэтому целевой функцией для таких задач будут затраты на заработную плату:

$$f_2 = S_{estd}/x_{di} + S_{estq}/x_{qj} \rightarrow \min.$$

Получим математическую модель задачи:

$$\begin{cases} f_2 = S_{estd}/x_{di} + S_{estq}/x_{qj} \rightarrow \min, \\ S_{estd}/x_{di} + S_{estq}/x_{qj} \leq S_{est} \\ x_{di} > 0 \\ y_{qj} > 0 \end{cases} \quad (2)$$

где S_{est} – оценочный общий бюджет на решение задачи;

S_{estd} – оценочный бюджет, выделенный на разработку;

S_{estq} – оценочный бюджет, выделенный на тестирование.

Указанные выше коэффициенты эффективности каждого из работников для задач типа *minor* рассчитываются на основании истории ранее выполненных задач, аналогично табл. 1, 2.

Полученные математические модели (1) и (2) являются задачами нелинейного программирования с одним критерием оптимальности, для которых существуют стандартные методы решения [6]. Проведенные предварительные расчеты на тестовом наборе данных свидетельствуют об эффективности предложенного подхода.

Таким образом, может быть получена возможность минимизации затрат на каждой задаче типа *minor* и уменьшение времени на процесс разработки задач типа *critical*. Придерживаясь данного алгоритма, можно автоматизировать процесс назначения работников на задачи, а также оптимизировать расходы ИТ-предприятия. Предложенная методика может быть расширена за счет рассмотрения других видов ресурсов, используемых при работе над ИТ-проектом.

Математическая модель распределения трудовых ресурсов реализовывалась при разработке программно-методического комплекса для управления ИТ-проектами [7], диаграмма прецедентов бизнес-процесса которого представлена на рис. 1.

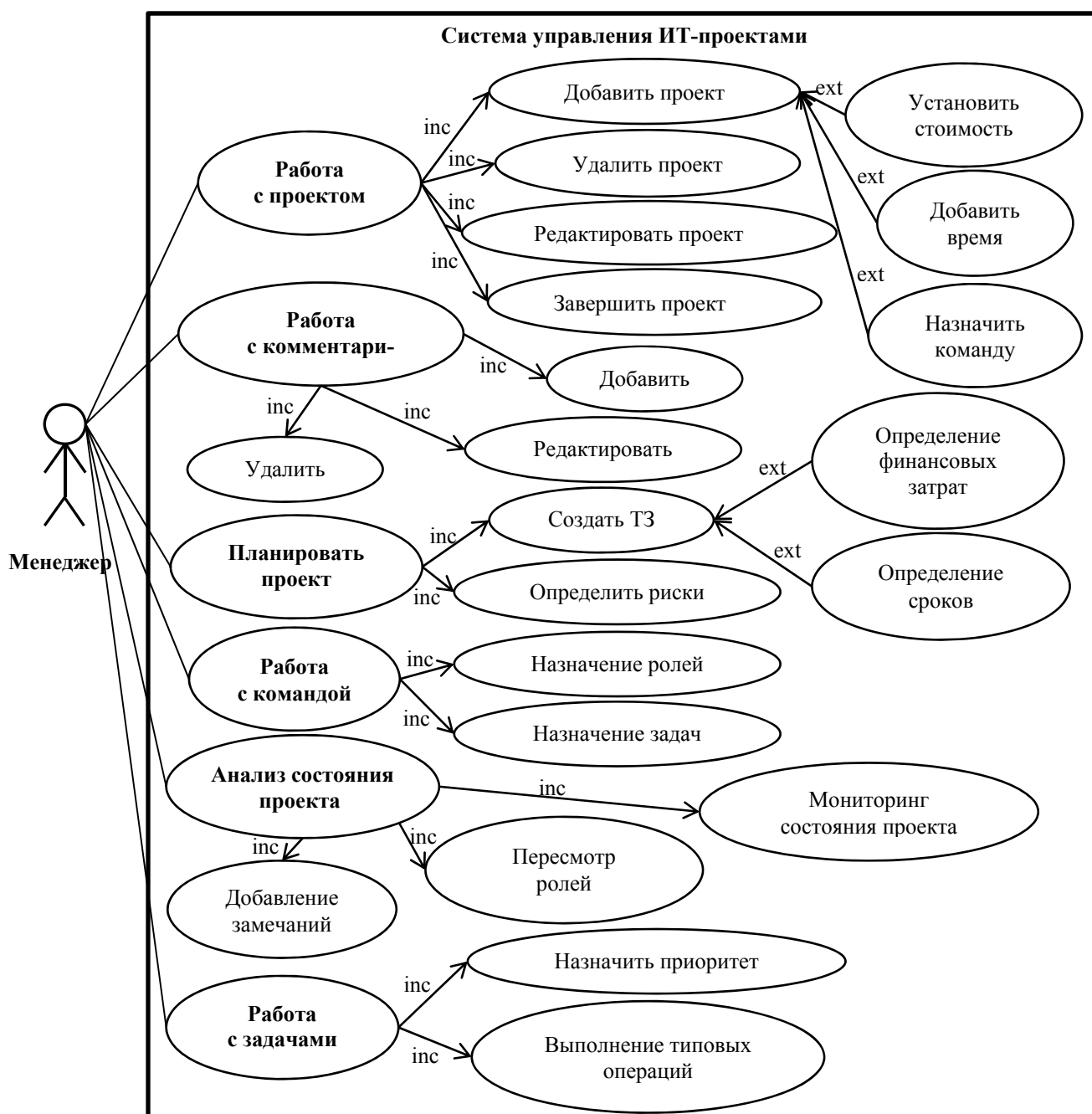


Рис. 1. Диаграмма прецедентов бизнес-процесса «Система управления ИТ-проектами»

ВЫВОДЫ

В проведенном исследовании были проанализированы возможности управления трудовыми ресурсами с учетом их квалификации при их распределении на проекты (задачи). Построены оптимизационные математические модели распределения трудовых ресурсов для двух типов задач. Предлагаемые модели позволяют учитывать особенности проекта и его участников. Использование разработанных моделей в задачах по распределению или перераспределению работников по проектам позволяет снизить нагрузку на менеджера, ускорить процесс принятия решения. Модели для оптимизации состава команды IT-проекта могут быть использованы для снижения рисков за счет предварительного планирования.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Ломазов В. А. Поддержка принятия решений при оценивании IT-проектов / В. А. Ломазов, В. И. Ломазова, В. С. Нехотина // *Международный журнал прикладных и фундаментальных исследований*. – 2015. – №. 3–2. – С. 170–173.
2. Wrike [Электронный ресурс]. – Режим доступа: <https://startpack.ru/application/wrike-project-management>.
3. Jira [Электронный ресурс]. – Режим доступа: <https://startpack.ru/application/jira>. Дата обращения: 5.04.2017.
4. Шафер Д. Управление программными проектами: достижение оптимального качества при минимуме затрат [Текст] / Д. Шафер, Р. Фатрелл, Л. Шафер. – М. : Вильямс, 2003. – 1136 с.
5. Сланик В. С. Информационные системы поддержки принятия решений в управлении IT-проектами [Электронный ресурс] / В. С. Сланик. – Режим доступа: <http://elib.bsu.by/handle/123456789/52171>.
6. Васильева Л. В. Використання комп'ютерних технологій для розв'язання оптимізаційних задач в економіці : навч. посібник / Л. В. Васильева, І. А. Гетьман. – Краматорськ : ДДМА, 2011. – 200 с.
7. Шелест А. И. Разработка программного комплекса для управления IT-проектами на предприятии среднего бизнеса / А. И. Шелест, Л. В. Васильева // *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку : матеріали Всеукраїнської науково-практичної Internet-конференції*. – Черкаси, 2017. – С. 101-102. Режим доступа: <https://conference.ikto.net/public/accepted.html>.

Статья поступила в редакцию 08.10.2017 г.