

Ю. В. Вайнилович, старший преподаватель, e-mail: Ylia.v@tut.by,
Межгосударственное образовательное учреждение высшего образования
"Белорусско-Российский университет", г. Могилев, Республика Беларусь

Метод повышения эффективности управления IT-проектами с использованием генетического алгоритма

Рассматривается проблема повышения эффективности управления IT-проектами. Предлагается метод управления IT-проектами с использованием генетического алгоритма. Предложенный метод отличается от существующих подходом к формированию проектных команд и распределением участников команды на задачи проекта.

Ключевые слова: управление IT-проектами, проектная команда, генетический алгоритм

Введение

Бурное развитие новых технологий, возрастающая конкуренция на рынке программных продуктов дают заказчикам возможность предъявлять все более высокие требования как к качеству программных продуктов, так и к сокращению бюджетов и сроков на разработку. Эта тенденция обусловила потребность в новых методах управления.

Д. А. Новиков, Е. В. Коновальчук занимались разработкой и исследованием методов управления проектами в процессе их реализации с учетом изменившихся условий [1, 2]. Темой исследований В. Н. Буркова является специфика человека как объекта и субъекта управления [3]. Научные исследования в области управления проектами проводились В. И. Воропаевым (разработаны обобщенные сетевые модели) [4], Б. П. Титаренко (создание робастной технологии) [5], В. М. Аньшиным (формирование и управление портфелем проектов) [6].

Наиболее распространенными методами управления IT-проектом являются методология Agile, проектное управление с использованием метода критической цепи, Critical Path Method (CPM) — метод критического пути, XP-методы, Scrum, Kanban, поточный метод планирования (водопадная модель) [1].

При использовании перечисленных методов управления проектами менеджеры сталкиваются с проблемой подбора членов проектных команд, распределения задач по исполнителям,

планирования выполнения проекта после его начала при внесении корректировок в первоначальный план.

Отсутствие необходимых методик и инструментов не позволяет менеджеру проектов адекватно оценить компетенции и навыки участника проектов, его личностные качества и факторы мотивации, что приводит к превышению сроков выполнения проекта, сдаче проекта, в котором часть требований заказчика не была реализована.

В связи с этим актуальной является задача повышения эффективности управления IT-проектами.

В данной статье предложен метод повышения эффективности управления IT-проектом, позволяющий снизить трудоемкость решения задач и уменьшить время выполнения проекта. Расчет трудоемкости и времени выполнения проекта осуществляется по методике СОСОМО II (подробнее об этой методике см. в работе [7]):

$$PM = \sum_{k=1}^N \left(PM_k^B \cdot \prod_{i=1}^6 EM_i \right) \rightarrow \min,$$

где PM — трудоемкость проекта; EM_i — множители трудоемкости ($i = 1, \dots, 6$); PM_k^B — базовая трудоемкость k -й задачи проекта в чел.мес; N — число задач проекта;

$$TDEV = C(PM_{NS})^{D+0,2 \cdot 0,01 \cdot \sum_{j=1}^5 SF_j} \cdot \frac{SCED}{100} \rightarrow \min,$$

где $TDEV$ — длительность проекта; C, D — коэффициенты, определенные разработчиками модели, калибруемые по статистическим данным, по калибровке 2000 г. $C = 3,67, D = 0,28$; PM_{NS} — трудоемкость проекта без учета множителя $SCED$, определяющего сжатие расписания; SF_j — факторы масштаба ($j = 1, \dots, 5$).

Исходными данными для решения задачи являются компоненты ИТ-проектов — проектная команда, множество задач, приоритет проектов, трудоемкость проекта, длительность проекта:

$$IT = \langle Team, Task, Priority, PM, TDEV \rangle, \quad (1)$$

где $Team$ — проектная команда; $Task$ — множество задач; $Priority$ — приоритет проекта.

Каждый участник проектов характеризуется определенным набором параметров:

$$TP = \langle PPQ, Tech, LTech, Ex \rangle, \quad (2)$$

где PPQ — набор параметров, характеризующих личностно-психологические качества участников проектов; $Tech$ — перечень технологий, которыми владеет участник проектов; $LTech$ — уровень владения технологиями; Ex — опыт участия в проектах с использованием технологии.

Каждая задача проекта из множества $Task$ задается набором характеристик:

$$Task = \langle T, Type, TTech, Priority \rangle, \quad (3)$$

где T — время выполнения задачи; $Type$ — тип задачи; $TTech$ — применяемая технология; $Priority$ — приоритет задачи.

Отличительной особенностью предложенного метода является использование генетического алгоритма [8—10], который при построении проектной команды и планировании выполнения проекта учитывает личностно-психологические качества, уровень владения технологиями, опыт работы с технологиями, сработанность участников проектной команды.

Разработанный метод позволяет сократить время выполнения проекта за счет более качественного подбора проектной команды и планирования выполнения проекта при динамически изменяющихся требованиях и целях.

Метод состоит из следующих шагов, описанных ниже:

- 1) оценка участников проектов;
- 2) формирование проектной команды;
- 3) планирование выполнения проекта.

Шаг 1. Оценка участников проектов

В результате этого шага определяется набор параметров, характеризующих личностно-психологические качества участников проектов, предпочитаемую роль в команде [11, 12], уровень владения технологиями, опыт применения технологий при реализации проектов.

Исследование личностно-психологических особенностей участников проектов. Исследование проводится на основе автоматизированных тестов. Набор тестов включает тесты Лири, Белбина, Майерс—Бриггс, Томаса, Хони—Мамфорда. Результаты тестов позволяют определить тип личности, стиль поведения людей в конфликтных ситуациях, склонность к виду деятельности [13—17], стиль обучения [18], командные роли, которые может исполнять участник проектов.

Каждый тест описывается набором признаков, состоящих из множества вопросов тестовых заданий, множества ответов объекта исследования, множества результатов обработки ответов, интерпретации результатов теста.

Оценка взаимосвязи показателей, полученных в результате тестирования, проводится методом кластерного анализа k -средних на этапе формирования команды для выполнения проекта. В результате все претенденты на включение в проектную команду должны быть разделены на кластеры (группы) с учетом их интеллектуально-личностного потенциала.

Оценка уровня владения технологиями и инструментами и опыта работы с технологиями и инструментами. Оценка формируется на основании результатов участия в предыдущих проектах.

Перечень технологий $Tech$, которыми владеет участник проектной группы, формируется из множества T технологий и инструментов.

Множество T содержит перечень всех технологий и инструментов, применяемых для разработки ИТ-проектов.

Для каждой технологии из множества T оценивается уровень владения и опыт разработок участника проектов:

$$LTech = \langle LT, Ex \rangle,$$

где LT — оценка уровня владения технологией; Ex — опыт участия в проектах с использованием данной технологии.

Для системных аналитиков, проектировщиков баз данных, программистов уровень владения технологией вычисляется для каждого завершенного проекта по формуле

Оценка опыта работы с технологиями и инструментами

Супернизкий	Очень низкий	Низкий	Нормальный	Высокий	Очень высокий	Супервысокий
Менее 1500	1500...3000	3001...4500	4501...6000	6001...7500	7501...9000	10000 и более

$$LT = \frac{\sum_{i=1}^{N\text{Tasks}} \text{quality}_i + \sum_{i=1}^{N\text{Tasks}} \text{speed}_i}{N\text{Tasks}} \cdot \text{complexity}.$$

Здесь quality_i — качество решения i -й задачи проекта, которое находится по формуле

$$\text{quality} = 1 - \frac{\text{ConfirmedErrors}}{\text{FindErrors}},$$

где ConfirmedErrors — число подтвержденных ошибок; FindErrors — число найденных ошибок; speed — скорость решения задачи проекта, которая находится как отношение планового срока решения задачи к фактическому сроку решения:

$$\text{speed} = \frac{T\text{Planned}}{T\text{Actual}},$$

где $T\text{Planned}$ — плановое время решения задачи; $T\text{Actual}$ — фактическое время решения задачи; $N\text{tasks}$ — число задач проекта, решенных участником; complexity — сложность задачи.

Сложность задачи определяется по табл. 1 отраслевого стандарта 4.071.030 "Создание системы. Нормативы трудоемкости" [19].

Для тестировщиков уровень владения технологиями оценивается по формуле

$$\text{quality} = 1 - \frac{\text{NoErrors}}{\text{FindErrors}},$$

где NoErrors — число неподтвержденных ошибок.

Опыт работы с технологиями и инструментами участника проектов Experience определяется по правилу 10 000 часов [20]. Для этого находится суммарная длительность всех проектов с использованием оцениваемой технологии и в соответствии с табл. 1 определяется уровень владения.

Таким образом, в результате прохождения первого этапа каждый участник проектов характеризуется набором параметров (2).

Шаг 2. Формирование команд

Предварительная оценка трудоемкости проекта. Предварительная оценка трудоемкости и

длительности проекта осуществляется менеджером проекта по методике COCOMO II [7].

Для оценки трудоемкости проекта SCRUM-мастер разбивает проект на задачи и оценивает объем программного кода каждой задачи в тысячах строк исходного текста.

На данном шаге все значения множителей трудоемкости оцениваются как "нормальный", значения факторов масштаба — как "средний".

Распределение участников проектов в проектные команды. IT-проект состоит из компонентов (1).

Компонент Team имеет сложную структуру. Его можно декомпозировать следующим образом:

$$\text{Team} = \langle \text{Role}, \text{TP} \rangle,$$

где Role — множество командных ролей; TP — множество участников проектной команды.

Задача формирования команды проекта состоит в том, чтобы найти такой вариант распределения, при котором трудоемкость проекта и срок выполнения проекта минимизируются. При этом выполняются следующие условия:

- в одной команде окажутся психологически совместимые участники;
- на каждую командную роль будет назначен участник с предпочитаемой ролью;
- уровень владения требуемыми технологиями участниками проектной команды — максимальный.

Таким образом, сформулирована задача оптимизации: сформировать такую проектную команду Team из множества возможных разбиений, при которой

$$\begin{aligned} PM(\text{Team}) &\rightarrow \min; \\ TDEV(\text{Team}) &\rightarrow \min. \end{aligned}$$

Для решения задачи применим генетический алгоритм [21, 22].

Шаг 1. Первоначальное создание группы проектных команд. задается population_{\max} — максимальное число проектных команд, существующих одновременно; N_{step} — число шагов, $i = 1$ — номер итерации; m — число команд, формируемых на первом шаге.

Каждая проектная команда формируется в соответствии с определенными правилами

Уровни множителя трудоемкости PERS по методике СОСОМОИ

Супернизкий	Очень низкий	Низкий	Нормальный	Высокий	Очень высокий
$\leq 0,143$	0,144...0,286	0,287...0,429	0,430...0,571	0,572...0,714	0,715...0,857

ми. На каждую командную роль выбирается участник проектов из числа нераспределенных, у которого

- данная роль является предпочитаемой;
- если участники с предпочитаемой ролью отсутствуют, то выбирается участник, у которого роль является поддерживающей;
- психологически совместим с ранее выбранными участниками команды;
- владеет хотя бы одной, требуемой для реализации проекта, технологией.

Создание первоначального набора проектных команд завершается в следующих случаях:

- все участники проектов распределены по командам;
- отсутствуют участники с предпочитаемыми и поддерживающими ролями для назначения на командную роль.

Далее вычисляется функция приспособленности для каждой команды — длительность и время выполнения проекта.

Рассчитывается уровень множителя трудоемкости PERS — квалификация команды. Для этого проводится нормирование уровня владения технологиями каждого участника проектной команды

$$LTech_{team}^{norm} = \frac{LTech_{team}}{LTech_{max}}.$$

Уровень владения технологиями командой определяется по формуле

$$LTech_{general} = \frac{\sum LTech_{team}^{norm}}{N},$$

где N — число участников проектной команды.

Уровень множителя трудоемкости PERS выбирается в соответствии с табл. 2.

Рассчитывается уровень множителя PREX — опыт персонала — как среднее значение по команде.

Рассчитывается уровень факторов масштаба PREC — наличие опыта аналогичных разработок.

Рассчитывается уровень фактора масштаба TEAM — сработанность команды. Для этого вычисляется наибольший процент членов команды, принимавших участие в разработке

Таблица 3

Уровни фактора масштаба TEAM по методике СОСОМОИ

Очень низкий	Низкий	Средний	Высокий	Очень высокий	Критический
20 %	40 %	60 %	75 %	90 %	100 %

одного проекта. Уровень фактора выбирается в соответствии с табл. 3.

Рассчитываются PM — трудоемкость проекта и $TDEV$ — длительность выполнения проекта для каждой проектной команды.

Шаг 2. Селекция — отбор проектных команд для формирования на их основе новых команд. Будем использовать селективный отбор, при котором следует выбирать только те команды, значения функции приспособленности которых не больше пороговой величины — среднего значения приспособленности по всем командам

$$PM_{it}^k \leq \frac{\sum_{k=1}^m PM_{it}^k}{m}.$$

Результатом этого шага будут две команды s^1 и s^2 .

Шаг 3. Скрещивание. Для формирования новых команд используется одноточечное скрещивание — выбирается точка разрыва, обе команды делятся на две части по этой точке и меняются соответствующими частями.

Точка разрыва показывает, сколько членов команды окажутся в первом сегменте, сколько — во втором. Точка разрыва выбирается случайным образом из промежутка $[1, N]$ (где N — число участников проектной команды).

Результатом шага 3 являются две новые команды s^{ch1} и s^{ch2} , полученные путем применения оператора скрещивания к исходным командам s^1 и s^2 .

Шаг 4. Мутация — изменение структуры команды, случайно меняющее одного (или несколько) ее участников. Мутация в рамках решаемой задачи заключается в изменении ролей для двух участников команды.

Варианты обмена:

1. В одной команде имеется участник, назначенный на предпочитаемую для него роль X и имеющий поддерживающую роль Y , во вто-

рой команде имеется участник, назначенный на роль Y , которая является для него поддерживающей и имеющий предпочитаемую роль X — эти участники обмениваются ролями.

2. В команде имеется участник, назначенный на неподходящую для него роль X и имеющий предпочитаемую (или поддерживающую) роль Y , во второй команде имеется участник, назначенный на предпочитаемую (или поддерживающую) для него роль Y — эти участники могут обмениваться ролями (например, если уровень владения технологиями второго участника в данной роли выше, чем у первого).

Результатом шага 4 являются две новые команды s^{M1} и s^{M2} , полученные путем применения оператора мутации к исходным командам s^1 и s^2 , либо к командам, полученным на шаге 3 s^{ch1} и s^{ch2} .

Шаг 5. Сокращение промежуточного числа проектных команд. Применяется стратегия элитарного отбора. Создается промежуточный набор команд, который включает как исходные команды, так и команды, полученные в результате селекции и мутации. Для всех команд вычисляется функция приспособленности, а затем из них выбираются $population_{max}$ самых лучших.

Проверяются условия:

- если $i < N_{step}$, то полагаем $i = i + 1$ и переходим к шагу 2;
- если $i = N_{step}$, то переходим к шагу 6.

Шаг 6. Окончание работы генетического алгоритма. В качестве решения задачи выбирается проектная команда с наилучшим значением функции приспособленности.

Шаг 3. Планирование выполнения проекта

Как было показано выше, каждая задача проекта (3) характеризуется следующими параметрами: трудоемкость, тип, используемая технология, приоритет

$$Task = \langle T, Type, Tech, Priority \rangle.$$

Трудоемкость каждой задачи проекта определена на этапе формирования команды.

Тип и используемая технология определяют на этапе формирования списка задач проекта.

Приоритет задачам присваивает владелец программного продукта.

На первом этапе планирования каким-либо методом, например методом сетевого планирования, формируется первоначальный план выполнения проекта.

Далее формируется выборка задач, которые имеют наивысший приоритет, могут выполняться одновременно и не зависят от завершения других задач:

$$SelectingTask = \{Task_1, Task_2, \dots, Task_n\}.$$

Затем на каждую задачу должен быть назначен исполнитель из числа участников проектной команды:

$$SelectingTeam = \{Team_1, Team_1, \dots, Team_n\}.$$

Оптимальным будет считаться такое распределение исполнителей, при котором для сформированной выборки задач проекта

$$PM_{SelectingTask}(SelectingTeam) \rightarrow \min;$$

$$TDEV_{SelectingTask}(SelectingTeam) \rightarrow \min.$$

Если объем выборки оказался больше числа членов проектной команды, то к исполнению проекта можно привлечь свободных исполнителей или участников других проектов с недостаточной загрузкой. Если привлечь дополнительных исполнителей нет возможности, то следует сократить выборку задач до числа участников проектной команды.

Для назначения исполнителей на задачи применим генетический алгоритм [21, 22].

Шаг 1. Первоначальное распределение исполнителей на задачи. Задается $population_{max}$ — максимальное число вариантов распределения, существующих одновременно; N_{step} — число шагов; $i = 1$ — номер итерации; m — число вариантов распределения, формируемых на первом шаге. Варианты распределения формируются случайным образом с учетом следующих ограничений:

- каждый вариант распределения содержит число исполнителей, равное числу задач;
- один и тот же исполнитель может присутствовать в нескольких вариантах распределения;
- один и тот же исполнитель не может присутствовать в одном варианте распределения более одного раза;
- варианты распределения, отличающиеся только порядком следования исполнителей, считаются различными.

Будем считать, что исполнитель, стоящий в варианте распределения на первом месте, назначен на исполнение первой задачи в выборке задач и т. д.

Далее вычисляется функция приспособленности для каждого варианта распределения.

Рассчитывается уровень множителя трудоемкости PERS — квалификация команды как среднее арифметическое показателей каждого участника команды.

При этом для каждого участника команды учитывается уровень владения только той технологией, которая необходима для решения задачи, на которую он назначен.

Определение показателей остальных уровней и факторов масштаба, а также расчет трудоемкости и длительности выполнения выборки задач проекта осуществляется в соответствии с методикой распределения участников проектов по проектным командам.

Шаг 2. Селекция — отбор вариантов распределения для формирования новых вариантов. Будем использовать селективный отбор, при котором следует выбирать только те варианты распределения, значение функции приспособленности которых не больше пороговой величины — среднего значения приспособленности по всем вариантам

$$PM_{it}^k \leq \frac{\sum_{k=1}^m PM_{it}^k}{m}.$$

Результатом этого шага будут два варианта распределения s^1 и s^2 .

Шаг 3. Скрещивание. Для получения нового варианта распределения будем использовать однородный кроссинговер. Его суть состоит в том, что каждый исполнитель варианта распределения-потомка создается путем копирования соответствующего исполнителя из первого или второго варианта распределения-родителя. Для этого случайным образом генерируется двоичная маска кроссинговера той же длины (с тем числом битов), что и у вариантов распределения-родителей. Четность бита маски показывает родителя, из которого копируется исполнитель потомка (например, 1 соответствует первому родителю, а 0 — второму).

Новый вариант распределения, полученный по маске, должен удовлетворять ограничениям, перечисленным в шаге 1. Если какое-либо ограничение нарушается, следует сгенерировать новую двоичную маску.

Результатом шага 3 являются два варианта распределения-потомка s^{ch1} и s^{ch2} , полученные путем применения оператора скрещивания к вариантам распределения-родителям s^1 и s^2 .

Шаг 4. Мутация — изменение структуры варианта распределения. Мутация в рамках решаемой задачи заключается в изменении для

двух исполнителей варианта распределения решаемой задачи.

Результатом шага 4 являются новые варианты распределения s^{Mi} , полученные путем применения оператора мутации к участникам вариантов распределения s^{chi} .

Шаг 5. Сокращение промежуточной популяции вариантов распределения. Применяется стратегия элитарного отбора. Создается промежуточный набор вариантов распределения, который включает как исходные варианты распределения, так и варианты распределения, полученные в результате селекции и мутации. Для всех вариантов распределения высчитывается функция приспособленности, а затем из них выбираются $population_{max}$ самых лучших.

Проверяются условия:

- если $i < N_{step}$, то полагаем $i = i + 1$ и переходим к шагу 2;
- если $i = N_{step}$, то переходим к шагу 6.

Шаг 6. Окончание работы генетического алгоритма. В качестве решения задачи выбирается вариант распределения исполнителей на задачи с наилучшим значением функции приспособленности из последней популяции.

Апробация методики

Для апробации предложенного метода управления ИТ-проектами разработано программное обеспечение [23, 24].

Апробация проводилась в рамках учебной практики в группе студентов по специальности "Программная инженерия" Белорусско-Российского университета, которая состоит из пятнадцати человек ($P1, \dots, P15$), с уровнем владения требуемыми технологиями и опытом работы, представленными в табл. 4.

Для реализации был выбран проект со следующими характеристиками:

- требуется использовать пять технологий $T1, T2, T3, T4, T5$;
- объем программного кода по предварительным оценкам 8 тысяч строк.

Из группы студентов формировали три проектные команды, которым был предложен для реализации один и тот же проект.

Первая группа формировалась с помощью предложенного алгоритма, две другие — по желанию студентов.

По предварительной оценке трудоемкости и длительности проекта по методике СОСОМО II $PM = 24,12$ чел./мес, $TDEV = 0,10$ мес.

Шаг 1. Первоначальное создание группы проектных команд. Максимальное число про-

Таблица 4

Характеристики участников проекта

Участники проектов	Уровень владения технологиями					Опыт персонала				
	T1	T2	T3	T4	T5	T1	T2	T3	T4	T5
P1	0,12	0,64	0,13	0,81	0,06	7	2	2	7	3
P2	0,00	0,63	0,41	0,95	0,36	0	2	7	6	5
P3	0,93	0,06	0,37	0,56	0,39	4	1	2	1	5
P4	0,61	0,21	0,89	0,23	0,09	6	3	6	2	5
P5	0,71	1,00	0,51	0,35	0,90	6	6	1	4	1
P6	0,45	0,44	0,07	0,83	0,26	2	1	1	1	2
P7	0,37	0,50	0,62	0,21	0,35	1	6	5	5	6
P8	0,05	0,00	0,16	0,35	0,33	2	0	1	1	3
P9	0,69	0,65	0,60	0,94	0,30	3	4	7	1	6
P10	0,35	0,94	0,03	0,00	0,00	1	4	1	0	0
P11	0,53	0,08	0,26	0,00	0,61	6	7	1	0	2
P12	0,65	0,78	0,44	0,76	0,49	2	6	3	4	2
P13	0,43	0,50	0,70	0,24	0,61	2	5	3	6	4
P14	0,85	0,00	0,46	0,69	0,79	4	0	3	6	4
P15	0,44	0,93	0,38	0,80	0,01	6	6	7	4	2

ектных команд, существующих одновременно $population_{max} = 3$. Число шагов алгоритма $N_{step} = 3$. Считается, что все требования к формированию проектных команд выполнены. На первом шаге формируется три команды ($m = 3$)

$$Team1 = \{P1, P2, P3, P4, P5\};$$

$$Team2 = \{P10, P6, P7, P8, P9\};$$

$$Team3 = \{P11, P12, P13, P14, P15\}.$$

Рассчитывается трудоемкость и длительность выполнения проекта для каждой проектной команды:

$$PM_{team1} = 24,12; TDEV_{team1} = 0,10;$$

$$PM_{team2} = 30,88; TDEV_{team2} = 0,11;$$

$$PM_{team3} = 19,51; TDEV_{team3} = 0,09.$$

Шаг 2 (итерация 1). Селекция. Рассчитывается среднее значение приспособленности по всем командам

$$\frac{\sum_{k=1}^3 PM_{team}^k}{3} = 24,84.$$

Значения функции приспособленности не больше пороговой величины у команд *Team1* и *Team3*. Следовательно, эти команды будут участвовать в формировании новых команд.

Шаг 3 (итерация 1). Скрещивание. Случайным образом выбирается точка разрыва из диапазона [1, 5]. Пусть точка разрыва $BreakPoint = 3$. Проектные команды обмениваются частями, стоящими после точки разрыва и образуют две новые команды:

$$Team4 = \{P1, P2, P3, P14, P15\};$$

$$Team5 = \{P11, P12, P13, P4, P5\}.$$

Шаг 4 (итерация 1). Мутация. Изменяются структуры команд *Team1* и *Team3* путем обмена участниками *P3* и *P15*:

$$Team6 = \{P1, P2, P15, P4, P5\};$$

$$Team7 = \{P11, P12, P13, P14, P3\}.$$

Шаг 5 (итерация 1). Сокращение промежуточного числа проектных команд. Рассчитываются трудоемкость и длительность выполнения проекта для проектных команд, полученных на шагах 3 и 4:

$$PM_{team4} = 23,50; TDEV_{team4} = 0,10;$$

$$PM_{team5} = 24,67; TDEV_{team5} = 0,10;$$

$$PM_{team6} = 23,57; TDEV_{team6} = 0,10;$$

$$PM_{team7} = 28,67; TDEV_{team7} = 0,11.$$

Из сформированных команд выбирается три ($population_{max} = 3$) с наилучшими показателями функции приспособленности — *Team3*, *Team4* и *Team6*.

Осуществляется переход к шагу 2 (итерация 2).

Результаты шагов итерации 2 представлены в табл. 5.

Шаг 6. Окончание работы генетического алгоритма. По результатам шага 5 итерации 3 лучшей командой является $Team10 = \{P3, P12, P13, P14, P15\}$.

Таким образом, для реализации проекта были сформированы следующие три команды:

- по результатам работы генетического алгоритма

$$Team10 = \{P3, P12, P13, P14, P15\};$$

- по желанию студентов

$$Team13 = \{P11, P12, P13, P14, P15\},$$

$$Team14 = \{P11, P2, P4, P1, P5\}.$$

Таблица 5

Результаты работы генетического алгоритма

Номер итерации	Шаг	Команда	PM_{teami}	$TDEV_{teami}$
2	2	$Team3 = \{P11, P12, P13, P14, P15\}$	19,51	0,09
		$Team4 = \{P1, P2, P3, P14, P15\}$	23,50	0,10
2	3	$Team8 = \{P1, P12, P13, P14, P15\}$	19,59	0,09
		$Team9 = \{P11, P2, P3, P14, P15\}$	23,36	0,10
2	4	$Team10 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
		$Team11 = \{P11, P2, P1, P14, P15\}$	23,31	0,10
2	5	$Team3 = \{P11, P12, P13, P14, P15\}$	19,51	0,09
		$Team10 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
		$Team11 = \{P11, P2, P1, P14, P15\}$	23,31	0,10
3	2	$Team3 = \{P11, P12, P13, P14, P15\}$	19,51	0,09
		$Team10 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
3	3	$Team12 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
		$Team13 = \{P11, P12, P13, P14, P15\}$	19,51	0,09
3	4	—	—	—
3	5	$Team10 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
		$Team12 = \{P3, P12, P13, P14, P15\}$	19,49	0,09
		$Team3 = \{P11, P12, P13, P14, P15\}$	19,51	0,09

Далее планирование выполнения IT-проекта и назначение исполнителей на работы для проектной команды $Team10$ выполнялось в соответствии с разработанной методикой. Команды $Team13$ и $Team14$ планировали выполнение проекта самостоятельно.

Предположим, что требуется назначить исполнителей из числа членов проектной команды $Team10$ на задачи проекта с характеристиками, представленными в табл. 6.

Шаг 1. Первоначальное распределение исполнителей на задачи. Число шагов алгоритма $N_{step} = 3$. Формируются $m = 3$ варианта распределения участников проектной команды:

$$\begin{aligned} SelectingTeam1 &= \{P3, P12, P13\}; \\ SelectingTeam2 &= \{P3, P14, P15\}; \\ SelectingTeam3 &= \{P12, P14, P13\}. \end{aligned}$$

Таблица 6

Характеристики задач проекта

Задачи	$Task1$	$Task2$	$Task3$
Требуемая технология	$T1$	$T4$	$T2$
Тыс. строк программного кода	2,1	1,4	1,5

Рассчитываются трудоемкость и длительность выполнения задач для каждого варианта назначения исполнителей:

$$\begin{aligned} PM_{SelectionTeam1} &= 10,18; TDEV_{SelectionTeam1} = 0,08; \\ PM_{SelectionTeam2} &= 8,99; TDEV_{SelectionTeam2} = 0,07; \\ PM_{SelectionTeam3} &= 8,90; TDEV_{SelectionTeam3} = 0,07. \end{aligned}$$

Шаг 2 (итерация 1). Селекция. Рассчитывается среднее значение приспособленности по всем вариантам распределения

$$\frac{\sum_{k=1}^3 PM_{it}^k}{3} = 9,36.$$

Значения функции приспособленности не больше пороговой величины у вариантов распределения $SelectionTeam2$ и $SelectionTeam3$. Следовательно, эти варианты распределения будут участвовать в формировании новых команд.

Шаг 3 (итерация 1). Скрещивание. Пусть двоичная маска кроссинговера имеет вид $Mask = \{1,0,0\}$. Получаем новые варианты распределения:

$$\begin{aligned} SelectingTeam4 &= \{P12, P14, P15\}; \\ SelectingTeam5 &= \{P3, P14, P13\}. \end{aligned}$$

Шаг 4 (итерация 1). Мутация. Проводится замена исполнителей задач внутри вариантов распределения $SelectionTeam2$ и $SelectionTeam3$:

$$\begin{aligned} SelectingTeam6 &= \{P15, P14, P3\}; \\ SelectingTeam7 &= \{P14, P12, P13\}. \end{aligned}$$

Шаг 5 (итерация 1). Сокращение промежуточной популяции вариантов распределения. Рассчитываются трудоемкость и длительность выполнения проекта для каждой проектной команды:

$$\begin{aligned} PM_{Selectionteam4} &= 7,88; TDEV_{Selectionteam1} = 0,07; \\ PM_{Selectionteam5} &= 10,20; TDEV_{Selectionteam2} = 0,08; \\ PM_{Selectionteam6} &= 13,65; TDEV_{Selectionteam3} = 0,08; \\ PM_{Selectionteam7} &= 11,73; TDEV_{Selectionteam3} = 0,08. \end{aligned}$$

Из исходных и сформированных вариантов распределения выбираются три ($population_{max} = 3$) с наилучшими показателями функции приспособленности — $SelectionTeam2$, $SelectionTeam3$ и $SelectionTeam4$.

Аналогичным образом проходятся итерации 2 и 3.

Шаг 6. Окончание работы генетического алгоритма. В качестве решения был выбран вариант распределения *SelectionTeam4*, так как он имеет наилучшее значение функции приспособленности.

Теоретический расчет на основе предложенного метода показал сокращение трудоемкости выполнения проекта с 24,12 чел./мес до 19,49 чел./мес. Эксперимент в образовательном процессе Белорусско-Российского университета подтвердил соответствие результатов теоретического расчета результатам, полученным на практике.

Заключение

В статье предложен метод повышения эффективности управления ИТ-проектами.

Предложенный метод отличается формированием проектных команд и распределением участников проектных команд для выполнения задач проекта на основе генетического алгоритма с учетом уровня владения технологиями, опыта работы с технологиями, сработанности участников проектной команды, опыта аналогичных разработок, личностно-психологических качеств участников проектов.

Критерием эффективности является минимизация трудоемкости и длительности выполнения проекта. В результате применения разработанного метода при проведении учебной практики студентов специальности "Программная инженерия" в Белорусско-Российском университете трудоемкость проекта сократилась на 19 %.

По результатам апробации разработанного метода выявлено следующее ограничение: для того чтобы оценить трудоемкость проекта, необходимо знать размер программного продукта в тысячах строк исходного кода, который неизвестен до окончания проекта. Следовательно, метод может иметь большую погрешность из-за неточности оценки размера программного продукта. Существенное влияние на результат применения метода оказывает человеческий фактор: психологическое состояние участников проекта, самочувствие.

Список литературы

1. Новиков Д. А. Управление проектами: организационные механизмы. М.: ПМСОФТ, 2007. 140 с.
2. Коновальчук Е. В., Новиков Д. А. Модели и методы оперативного управления проектами. М.: ИПУ РАН, 2004. 63 с.
3. Бурков В. Н., Коргин Н. А., Новиков Д. А. Как управлять проектами. М.: Либроком, 2009. 264 с.
4. Баркалов С. А., Воропаев В. И., Секлетова Г. И. и др. Математические основы управления проектами: учеб. пособие / Под ред. В. Н. Буркова. М.: Высшая школа, 2005.
5. Титаренко Б. П. Робастные технологии управления в ситуации неопределенности // Вестник МГАДА. Сер. Экономика. 2012. № 6. С. 119—121.
6. Аньшин В. М., Ильина О. Н. Исследование методологии оценки и анализ зрелости управления портфелями проектов в российских компаниях. М.: ИНФРА-М, 2018. 200 с.
7. Boehm V. et al. Software cost estimation with COCOMO II. Englewood Cliffs, NJ: Prentice—Hall, 2000.
8. Barricelli N. A. Symbiogenetic evolution processes realized by artificial methods // *Methodos*. 1957. N. 9 (35—36). P. 143—182.
9. Barricelli N. A. Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life // *Acta Biotheoretica*. 1963. N. 16. P. 99—126.
10. Melanie Mitchell. An Introduction to Genetic Algorithms. MIT Press, 1998. P. 167—226.
11. Белбин Р. М. Типы ролей в командах менеджеров / Пер. с англ. М.: НИРО, 2003. 232 с.
12. Белбин Р. М. Команды менеджеров. М.: Манн, Иванов и Фербер, 2009. 238 с.
12. Адакин Е. Е., Скрипникова Г. В. Оценка профессиональных квалификаций работников // *Фундаментальные исследования*. 2016. № 7-2. С. 282—286.
14. Бодров В. А. Психология профессиональной деятельности. М.: Институт психологии СО РАН, 2006. 623 с.
15. Ильин Е. П. Психология индивидуальных различий. СПб: Питер, 2011. 512 с.
16. Пряжников Н. С. Профессиональное самоопределение: теория и практика. М.: Академия, 2008. 320 с.
17. Шадриков В. Д. Профессиональные способности. М.: Университетская книга, 2010. 320 с.
18. Кудрявцев В. Т. Теория и практика учебной деятельности: традиции и инновации // *Психологическая наука и образование*. 2015. Т. 20, № 3. С. 197—218.
19. Отраслевой стандарт 4.071.030 "Создание систем. Нормативы трудоемкости". URL: <http://www.it-gost.ru/content/view/67/41> (дата обращения 25.03.2020).
20. Гладуэлл М. Гении и аутсайдеры: Почему одним все, а другим ничего? / Пер. с англ. О. Галкин. М.: Манн, Иванов и Фербер, 2016. 224 с.
21. Саймон Д. Алгоритмы эволюционной оптимизации. М.: ДМК Пресс, 2020. 940 с.
22. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы: Учеб. пособ. М.: Физматлит, 2006. 320 с.
23. Захарченков К. В., Вайнилович Ю. В. Методика многоуровневого управления учебными ИТ-проектами // *Энергетика, информатика, инновации — 2018 (инновационные технологии и оборудование в промышленности, управление инновациями, экономика и менеджмент, научные исследования в области физической культуры, спорта и общественных наук): сб. трудов VIII междунар. науч.-техн. конф.* Смоленск: "Универсум", 2018. Т. 3. С. 18—21.
24. Вайнилович Ю. В. Программный комплекс многоуровневого управления ИТ-проектами // *Инновации*. 2019. № 8(250). С. 88—96.

Method for Improving the Efficiency of IT Project Management Using a Genetic Algorithm

The article is devoted to solving the current problem of improving the efficiency of IT project management processes. When managing IT projects, managers are faced with the problem of formation teams and distributing tasks among project participants in the face of the need to minimize costs and completion dates of an IT project. The lack of necessary methods and software doesn't allow the IT project Manager to adequately assess competences and skills of participants, their personal qualities, which leads to a decrease in the effectiveness of project management. The article proposes a method of improving the efficiency of IT project management, which differs by using a genetic algorithm to form project commands and assign team participants to project tasks. The efficiency criterion is the complexity and duration of the project and individual tasks using the COCOMO II method. When forming project teams, takes into account the level of technologies proficiency, experience with technologies, the coherence of the project team members, and the experience of similar developments of project participants. The level of technologies proficiency affects the level of labor input multiplier, experience with technologies — at the level of the multiplier, the coherence of the project team members — on the level of scale factor, the experience of similar development — on the level of the scale factors of the COCOMO II methodology. Taking into account the personal and psychological qualities of project participants reduces the risk of interpersonal conflicts within the team, which also reduces the duration of projects and the labor input of solving tasks. Research of personal and psychological qualities is carried out on the basis of automated tests. The test suite includes Rosenzweig, Belbin, Myers-Briggs, Thomas and Honey-Mumford tests. The developed method is implemented in a software complex for multi-level IT project management. Testing of the method and software complex was carried out within the framework of the students' learning practice of the specialty "Software engineering" of the Belarusian-Russian University. The use of the proposed method allowed to reduce the labor input of solving the tasks of training projects by 19.2 %, to reduce the project realization term by 10 %.

Keywords: IT project management, project team, genetic algorithm

DOI: 10.17587/it.26.673-682

References

1. **Novikov D. A.** Project management: organizational mechanisms, Moscow, PMSOFT, 2007, 140 p. (in Russian).
2. **Konoval'chuk E. V., Novikov D. A.** Models and methods of operational project management, Moscow, IPU RAN, 2004, 63 p. (in Russian).
3. **Burkov V. N., Korgin N. A., Novikov D. A.** How to manage projects, Moscow, Librokom, 2009, 264 p. (in Russian).
4. **Barkalov S. A., Voropaev V. I., Sekletova G. I.** et al. Mathematical foundations of project management, Moscow, Vysshaya shkola, 2005, 423 p.
5. **Titarenko B. P.** Robust management technologies in a situation of uncertainty, *Vestnik MGADA, Seriya Ekonomika*, 2012, no. 6, pp. 119—121 (in Russian).
6. **An'shin V. M., Il'ina O. N.** Research of the assessment methodology and analysis of the maturity of project portfolios management in Russian companies, Moscow, INFRA-M, 2018, 200 p. (in Russian).
7. **Boehm B.** et al. "Software cost estimation with COCOMO II", Englewood Cliffs, NJ, Prentice—Hall, 2000.
8. **Barricelli N. A.** Symbiogenetic evolution processes realized by artificial methods, *Methodos*, 1957, no. 9(35—36), pp. 143—182.
9. **Barricelli N. A.** Numerical testing of evolution theories. Part II. Preliminary tests of performance, symbiogenesis and terrestrial life, *Acta Biotheoretica*, 1963, no. 16, pp. 99—126.
10. **Melanie Mitchell.** An Introduction to Genetic Algorithms, MIT Press, 1998, pp. 167—226.
11. **Belbin R. M.** Types of roles in management teams, Moscow, HIPPO, 2003, 232 p. (in Russian).
12. **Belbin R. M.** Management Teams, Moscow, Mann, Ivanov i Ferber, 2009, 238 p. (in Russian).
13. **Adakin E. E., Skripnikova G. V.** Assessment of professional qualifications of employees, *Fundamental'nye Issledovaniya*, 2016, no. 7-2, pp. 282—286 (in Russian).
14. **Bodrov V. A.** Psychology of professional activity, Moscow, Institut psikhologii SO RAN, 2006, 623 p. (in Russian).
15. **Il'in E. P.** Psychology of individual differences, Saint Petersburg, Piter, 2011, 512 p. (in Russian).
16. **Pryazhnikov N. S.** Professional self-determination: theory and practice. Moscow, Akademiya, 2008, 320 p. (in Russian).
17. **Shadrikov V. D.** Professional abilities, Moscow, Universitetskaya kniga. 2010, 320 p. (in Russian).
18. **Kudryavtsev V. T.** Theory and practice of educational activity: traditions and innovations, *Psikhologicheskaya Nauka i Obrazovanie*, 2015, vol. 20, no. 3, pp. 197—218 (in Russian).
19. **Industry** standard 4.071.030 "Creating a system. Labor intensity standards", available at: <http://www.it-gost.ru/content/view/67/41> (accessed March 25, 2020) (in Russian).
20. **Gladuell M.** Geniuses and outsiders: Why is everything for one and nothing for the other?, Moscow, Mann, Ivanov i Ferber, 2016, 224 p. (in Russian).
21. **Saymon D.** Evolutionary optimization algorithms, Moscow, DMK Press, 2020, 940 p. (in Russian).
22. **Gladkov L. A., Kureychik V. V., Kureychik V. M.** Genetic algorithms, Moscow, Fizmatlit, 2006. 320 p. (in Russian).
23. **Zakharchenkov K. V., Vaynilovich Yu. V.** Method for multi-level management of educational IT projects, *Energetika, informatika, innovatsii* — 2018 (innovatsionnye tekhnologii i oborudovanie v promyshlennosti, upravlenie innovatsiyami, ekonomika i menedzhment, nauchnye issledovaniya v oblasti fizicheskoy kul'tury, sporta i obshchestvennykh nauk): sb. trudov VIII mezhdunar. nauch.-tekhn. konf., Smolensk, Universum, 2018, vol 3, pp. 18—21 (in Russian).
24. **Vaynilovich Yu. V.** Program complex of multilevel IT project management, *Innovatsii*, 2019, no. 8 (250), pp. 88—96 (in Russian).