

Оптимизация управления ИТ-проектом с помощью моделирования его параметров в программе-симуляторе

Хитёв А. Ю.^{1, *}, Юрков А. В.²

¹ Санкт-Петербургский государственный университет, Санкт-Петербург, Российская Федерация; *alexkhitev@gmail.com

² Российский государственный педагогический университет им. А. И. Герцена, Санкт-Петербург, Российская Федерация

РЕФЕРАТ

Актуальными проблемами управления разработками ИТ-проектов является снижение нерационального расходования дорогостоящего времени специалистов, а также минимизация репутационных и иных потерь из-за непредвиденных отказов в обслуживании системы.

Целью исследований, которым посвящена статья, является количественный учет в модели оптимизации затрат основных процессов разработки ИТ-проекта.

На основе описанной в статье модели разработана программа-симулятор, ставшая инструментом для сравнительного анализа последствий возможных управленческих решений по оптимизации процесса разработки. В результате моделирования оказывается возможным количественно оценить совокупный эффект от различных изменений в организации работы над ИТ-проектом и понять, стоит ли идти на дополнительные затраты для подготовки и внедрения улучшений в процесс разработки.

Результаты вычислительных экспериментов на данных реальных ИТ-проектов позволили наглядно увидеть, что с экономической точки зрения предлагаемые управленческие решения ведут к снижению неоправданных затрат и минимизации возможных потерь. Как следствие, с использованием симулятора можно более осознанно отлаживать процесс реализации проекта и повысить предсказуемость работы команды и состояния проекта, а также оптимизировать использование ресурсов.

Как **вывод** выполненного исследования предложенная в статье модель реализации ИТ-проекта и разработанная на ее основе программа-симулятор выполнения проекта могут быть использованы для расчета эффекта от разнонаправленных оптимизаций с различной каденцией.

Исходные коды программы-симулятора открыты и доступны для скачивания, применения и адаптации.

Ключевые слова: программная инженерия, управление проектами, моделирование, симуляция, оптимизация, python, тестирование гипотез.

Для цитирования: Хитёв А. Ю., Юрков А. В. Оптимизация управления ИТ-проектом с помощью моделирования его параметров в программе-симуляторе // Управленческое консультирование. 2025. № 2. С. 93–109.

IT Project Management Optimization by Checking out of it's Parameters in the Simulation Application

Aleksei Yu. Khitev^{1, *}, Aleksandr V. Yurkov²

¹ Saint Petersburg State University, Saint Petersburg, Russian Federation; alexkhitev@gmail.com

² Herzen University, Saint Petersburg, Russian Federation

ABSTRACT

Current problems of IT project development management include reducing the irrational use of expensive specialist time, as well as minimizing reputational and other losses due to unexpected system service failures.

The aim of the research, which is devoted to the article, is the quantitative accounting in the cost optimization model of the main processes of IT project development.

Based on the model described in the article, a simulation program was developed, which became a tool for comparative analysis of the consequences of possible management decisions to optimize the development process. As a result of modeling, it becomes possible to quantitatively evaluate the cumulative effect of various changes in the organization of work on an IT project and understand whether it is worth incurring additional costs for the preparation and implementation of improvements in the development process.

The results of computational experiments on data from real IT projects made it possible to clearly see that, from an economic point of view, the proposed management decisions lead to a reduction in unjustified costs and a minimization of possible losses. As a result, using the simulator, you can more consciously debug the project implementation process and increase the predictability of the team's work and the state of the project, as well as optimize the use of resources.

As a conclusion of **the conducted** research, the model of IT project implementation proposed in the article and the project execution simulator program developed on its basis can be used to calculate the effect of multidirectional optimizations with different cadences.

The source codes of the simulator program are open and available for downloading, use and adaptation.

Keywords: software engineering, project management, simulation, modeling, optimization, python, hypothesis testing.

For citing: Khitev A. Yu., Yurkov A. V. IT Project Management Optimization by Checking out of it's Parameters in the Simulation Application // Administrative consulting. 2025. N 2. P. 93–109.

Введение

В классических работах Software Engineering Economics [10] и Abran A. Software project estimation [9] заложены основы экономических оценок для проектов программной инженерии. Однако есть ряд характерных особенностей управления современными программными проектами, которые не учитываются в классических схемах.

Работа команды над проектом представляет собой сложный клубок процессов, связей и мероприятий, происходящих с различной cadенцией¹. Исправление дефектов, подготовка к выпуску новых версий, реализация новых идей и устранение аварийных ситуаций — не полный их перечень.

При попытках внедрить небольшие изменения по различным направлениям возникает вопрос: как оценить совместное влияние возможных улучшений на весь процесс выполнения проекта в целом? Главную сложность вызывает различная частота улучшаемых процессов и изменение вероятности наступления событий. К примеру, вероятность наступления аварийной ситуации из-за определенного фактора снижается в связи с внедренным улучшением.

Цель работы

Целью отраженных в статье исследований стал количественный учет в модели оптимизации затрат основных процессов разработки ИТ-проекта. На основе разработанной модели для сравнительного анализа проведены вычислительные эксперименты в программе-симуляторе без улучшений и с улучшениями. В результате моделирования можно сравнить показатели и понять, стоит ли идти на дополнительные затраты для подготовки и внедрения улучшений.

¹ Cadенция в контексте разработки проекта означает ритм, длину итерации действия.

Моделированию проектов посвящен ряд публикаций. К примеру, в работе [11] автор строит модель для более точного определения требуемого времени выполнения, в статье [1] приводится общая модель функционирования организации и проектного подхода, в труде [6] представлена модель, учитывающая взаимодействие между членами команды. Однако авторам не удалось найти описания модели IT-проекта, учитывающей затраты времени и иных ресурсов на собрания, аварийные ситуации, поставки новых версий, а также расчет прибыли для организации.

Ниже в статье описана модель IT-проекта, которая легла в основу программы-симулятора выполнения проекта, рассчитывающая затраты и прибыль от разных процессов, учитывающая вероятность их наступления.

Задачи, решенные для достижения намеченной цели

1. Описание модели процесса разработки программного проекта.
2. Разработка и реализация программы-симулятора.
3. Оценка экономических преимуществ использования программы-симулятора.
4. Оценка перспектив развития созданного инструмента поддержки разработки программных проектов.

Актуальность работы

Нижеследующие исследования позволяют наглядно увидеть снижение затрат на нерациональное расходование дорогостоящего времени специалистов, а также репутационные и иные потери из-за непредвиденных отказов в обслуживании системы.

Кроме того, в результате использования описываемых методов, можно более осознанно отладить процесс работы проекта и повысить предсказуемость работы команды и состояния проекта и спрогнозировать оптимальное время утилизации ресурсов.

Методы для достижения намеченной цели

1. Моделирование процесса разработки программного проекта.
2. Программная реализация симулятора.
3. Вычислительные эксперименты с использованием данных реальных проектов.

Перечень условных обозначений, терминов и сокращений

- T_p — время, доступное для проекта за год.
 — время, выделяемое на итерацию.
 t_f — время, потраченное на разработку нового функционала.
 t_d — время, потраченное на исправление дефекта.
 t_{id} — время, потраченное на улучшение кода и работу с техническим долгом.
 t_{rm} — время, потраченное на регулярные совещания, включая ретроспективы, планирования, оценки сроков, ежедневный статус и прочие.
 \bar{t}_{rt} — среднее время, потраченное на развертывание новой версии, за год.
 \bar{t}_i — среднее время, потраченное на устранение аварии, за год.
 \bar{t}_{rdt} — среднее время отказа сервера в обслуживании из-за поставки новой версии.
 \bar{t}_{idt} — среднее время отказа сервера в обслуживании из-за аварии.

C_i — количество итераций за год.

C_{dt} — количество состояний отказа в обслуживании сервера за год.

C_t — количество аварийных ситуаций с состоянием отказа в обслуживании сервера за год.

C_r — количество поставок новой версии за год.

R_f — доход в единицу времени от разработанного нового функционала.

R_d — доход в единицу времени от устранения дефектов.

R_{td} — доход в единицу времени от устранения технического долга и улучшения кодовой базы.

$\sum_{D,T,BA,O,M} S$ — сумма зарплат в единицу времени специалистов, вовлеченных в активность, где D — программный инженер, T — инженер по тестированию, BA — бизнес-аналитик, O — системный инженер, M — руководитель.

E_p — расходы на проект в течение года.

E_o — расходы на решение обычных задач, таких как разработка нового функционала, исправление дефектов и закрытие технического долга в течение года.

E_s — затраты на работу с обычными задачами, такими как разработка нового функционала, исправление дефектов и закрытие технического долга в течение года.

E_r — затраты на поставку новой версии в течение года.

E_t — затраты на устранение аварийных ситуаций в течение года.

E_{dt} — потери из-за неработоспособности приложения.

E_o — прочие расходы, включая стоимость лицензий, серверов, инфраструктуры и прочего.

Описание модели проекта

Согласно Руководству PMBook [7], проект — это временное предприятие, направленное на создание уникального продукта, услуги или результата. Аналогичным образом, проект определяется в статье «Проектная модель функционирования организации» [1, с. 17] как временное (ограниченное по времени) предприятие, целью которого является создание услуги или товара, обеспечение определенного финансового или иного результата.

Однако проект еще можно представить как совокупность процессов, влияющих на объем функционала, качество продукта, срок достижения и прочие характеристики конечного результата в разработке. К таким процессам в разработке можно отнести:

- 1) процесс создания нового функционала,
- 2) процесс устранения дефектов и технического долга,
- 3) процесс подготовки версий,
- 4) процессы устранения аварийных ситуаций и выполнения срочных блокирующих заданий,
- 5) прочие регулярные активности, включая всевозможные совещания.

Как показано на рис. 1, разные процессы имеют разную каденцию (поставка новой версии) или вероятность наступления (возникновение аварии или срочной задачи).

Непосредственно в создании ценности участвуют следующие потоки.

1. Разработка нового функционала. За счет нового функционала увеличивается конверсия продаж, Retention Rate² и целесообразность увеличения стоимости продукта.

² Метрика, показывающая коэффициент удержания пользователей.

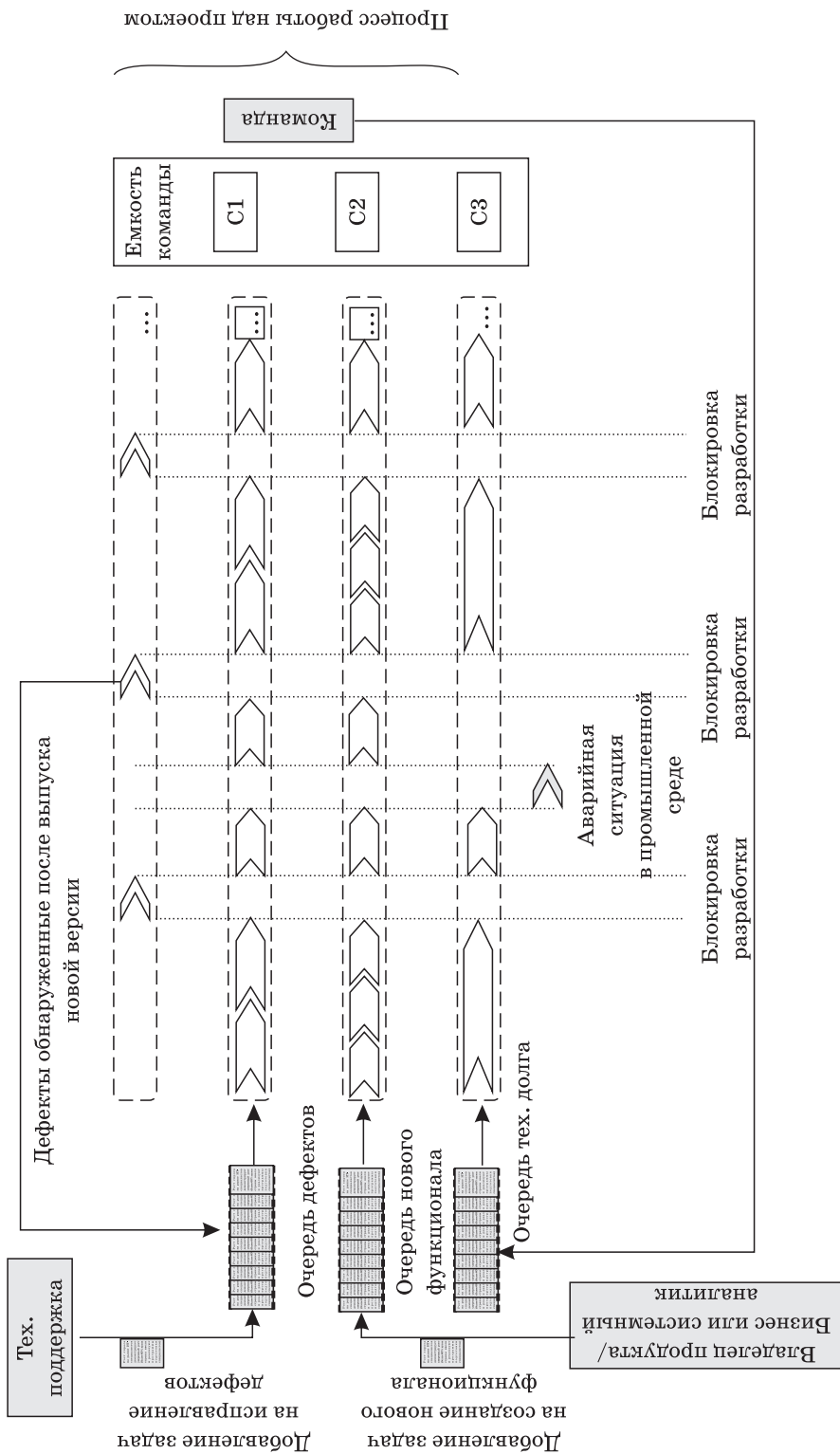


Рис. 1. Модель проекта в программной инженерии
Fig. 1. Software Project Model

Источник: [разработано авторами].

2. Устранение дефектов. Если пользователи видят, что постоянно ведутся работы по улучшению продукта и количество дефектов снижается, они, даже столкнувшись с дефектом, не будут сразу искать альтернативный продукт. Иными словами, снижается Churn Rate³.
3. Устранение технического долга. Поддержка хорошего качества архитектуры и кода продукта увеличивает скорость разработки нового функционала и исправление проблем, а также снижает вероятность возникновения дефектов и поддерживает стабильность работы команды, а значит, и соблюдение оценок выполнения задач.

Распределение времени в процессе выполнения проекта

В процессе выполнения проекта итерации имеют определенное доступное время, что означает, что объем работы, который может быть выполнен командой в течение итерации, ограничен. В случае возникновения аварии или срочной задачи время, которое может быть потрачено на работу, приносящую ценность, сокращается. К аналогичному сокращению времени приводит поставка новой версии и различные регулярные совещания.

На годовом интервале распределение времени можно представить таким образом:

$$T_p = (t_f + t_d + t_{id} + t_{rm})C_i + \bar{t}_{rt}C_r + \bar{t}_iC_i.$$

При условии, что

$$t_f + t_d + t_{id} + t_{rm} \leq t_i - \frac{\bar{t}_i * C_i}{C_i} - \frac{\bar{t}_{rt} * C_r}{C_i}.$$

Таким образом, время, потраченное на создание ценности в итерации, можно оценить, как

$$t_f + t_d + t_{id} = t_i - \frac{\bar{t}_i C_i}{C_i} - \frac{\bar{t}_{rt} * C_r}{C_i} - t_{rm}.$$

Доход от проекта

Зная время, потраченное на активности, создающие ценность, можно описать доход от проекта, как

$$t_f R_f + t_d R_d + t_{id} R_{id}.$$

Расходы на проект

Расходы на проект можно считать как сумму затрат на заработную плату специалистов, вовлеченных в разные активности, потери из-за неработоспособности приложения (из-за аварийной ситуации или поставки новой версии) и прочих расходов.

При этом годовые затраты на работу с обычными задачами, такими как разработка нового функционала, исправление дефектов и закрытие технического долга, можно представить, как:

$$E_s = \left(\sum_{D,T,BA,M} S * t_f + \sum_{D,T,BA,M} S * t_d + \sum_{D,T,M} S * t_{id} + \sum_{D,T,BA,O,M} S * t_{rm} \right) C_i.$$

³ Метрика, показывающая, какая доля пользователей перестала пользоваться продуктом.

Затраты на поставку новой версии, включая потери из-за неработающего приложения на время замены версии и зарплаты сотрудников, можно оценить, как:

$$E_r = \left(E_{dt} \bar{t}_{rdt} + \sum_{D,T,O,M} S * \bar{t}_{rt} \right) C_r .$$

Затраты на устранение аварийных ситуаций в течение года можно обозначить следующим образом:

$$E_t = \left(E_{dt} + \sum_{D,T,O,M} S \right) \bar{t}_{tdt} C_t .$$

На годовом интервале распределение времени можно представить таким образом:

$$E_p = E_s + E_r + E_t + E_o$$

при условии, что

$$t_f + t_d + t_{td} + t_{rm} \leq t_i - \frac{t_i C_i}{C_i} - \frac{t_{rt} * C_r}{C_i} .$$

Оптимизация проекта

Для максимизации создания ценности нужно:

- 1) сократить среднее время работы с аварийными ситуациями и количество аварийных ситуаций, что уменьшит простой приложения и увеличит среднюю емкость итерации, доступную для создания ценности;
- 2) уменьшить среднее время, требуемое для доставки новой версии приложения, что высвободит больше времени для выполнения специалистами задач, создающих ценность;
- 3) уменьшить или устранить время простоя при развертке приложения, что уменьшит издержки из-за простоя от неработоспособного приложения;
- 4) уменьшить среднее время, требуемое для работы над дефектом, что позволит либо устранять больше дефектов за то же время, либо перераспределить его в пользу создания нового функционала;
- 5) оптимизировать работу членов команды в целом.

Описание программы-симулятора проекта

Задача программы-симулятора состоит в том, чтобы оценить затраты на выполнение проекта и доходы от его реализации при разных входных условиях на большом количестве запусков.

Условия могут меняться в связи с применением того или иного средства оптимизации проекта. К примеру, применяя программу мониторинга системы.

Конфигурация программы-симулятора

Программа-симулятор конфигурируется в файле config.py⁴ в виде констант и ассоциативных массивов. Наиболее важные представлены в табл. 1.

⁴ https://github.com/aleksei-khitev/project_simulator/blob/master/config.py (дата обращения: 01.03.2025).

Описание части конфигурации
Table 1. Description of the configuration part

Константа	Описание
TEAM	Состав команды по ролям и количеству специалистов каждой роли
REGULAR_MEETINGS	Частота, продолжительность, состав участников различных совещаний
PROFIT_ACTIVITIES_AT_ITERATION_QUOTAS	Распределение времени, доступного для разработки, устранения дефектов и работы с техническим долгом
TROUBLES	Аварийные ситуации, включая вероятность их наступления, время и роли, требуемые для поиска причин проблемы и ее устранения

Листинг 1: Пример части конфигурации

```

TROUBLES = {
  "Service errors because of lack of server's disk space":
  {
    "time to find out root cause": {
      "time": 60, "roles": [SOFTWARE_ENGINEER, SYSTEM_
ENGINEER, QA_ENGINEER, PROJECT_MANAGER]
    },
    "time to recover": {
      "time": 15, "roles": [SOFTWARE_ENGINEER, SYSTEM_
ENGINEER]
    },
    "probability": 8 / WORKING_DAYS_IN_SPAIN
  },
  "Error because of some service down in the chain of
invocations": {
    "time to find out root cause": {
      "time": 15, "roles": [SOFTWARE_ENGINEER, SYSTEM_
ENGINEER, QA_ENGINEER, PROJECT_MANAGER]
    },
    "time to recover": {
      "time": 10, "roles": [SOFTWARE_ENGINEER, SYSTEM_
ENGINEER, QA_ENGINEER]
    },
    "probability": 20 / WORKING_DAYS_IN_SPAIN
  }
}

```

Описание модулей и их работы

В основном модуле main.py⁵ определена функция launch_simulations(simulation_count), которая «прогоняет» симуляцию с заданными параметрами указанное число раз, собирает и усредняет данные, полученные от запусков.

⁵ https://github.com/aleksei-khitev/project_simulator/blob/master/main.py (дата обращения: 01.03.2025).

В самом модуле:

- 1) готовятся данные для запусков;
- 2) вызывается описанная функция;
- 3) формируется отчет.

Установка параметров перед «прогонами» осуществляется за счет переопределения значений из конфигурации. Для формирования отчета вызываются функции из модуля `report.py`⁶. В итоге генерируется файл отчета в формате `markdown`, пример которого можно посмотреть в `report.md`⁷. Данный формат позволяет легко форматировать текст, описать формулы и таблицы. В дальнейшем файл в формате `markdown` не сложно преобразовать в формат `pdf` в большинстве редакторов (к примеру, в редакторе `MarkText`⁸). Пример результата представлен в репозитории в файле `report.pdf`⁹.

Модули `project.py`¹⁰ и `project_iteration.py`¹¹ содержат классы `Project` и `ProjectIteration` соответственно. В модуле `team.py`¹² содержится класс `Team`, который предназначен для вывода зарплат сотрудников, выплачиваемых для различных активностей в единицу времени.

Схематично данные классы изображены на рис. 2.

Класс `Project` содержит один публичный метод `play_year()`, который прогоняет нужное количество итераций проекта за год, в каждой итерации создает экземпляр класса `ProjectIteration`, вызывает его метод `play()` и собирает полученные данные в годовую статистику.

Класс `ProjectIteration` в методе `play()` вызывает приватные методы `__play_meetings()`, `__play_troubles()`, `__play_release()`, в ходе которых определяется, какие были совещания, аварии, была ли поставка новой версии, какие члены команды были заняты какими активностями и сколько это заняло времени. Далее рассчитываются затраты и время, потраченное на создание ценности. Эти данные в дальнейшем агрегируются сперва в `Project`, а потом и в модуле `main.py` для анализа и отчета.

Схематично последовательность изображена на рис. 3.

Описание реального проекта, который нуждался в оптимизации

Проект существовал уже несколько лет и имел устоявшиеся кодовую базу, процессы, пользователей. Его задачей было предоставление обработанных данных для биоинформатиков для построения экспериментов при работе над лекарственными препаратами.

Поставки новых версий происходили ежеквартально и были привязаны к поступлению большого объема данных из нескольких источников. Каждый релиз длился около двух дней и состоял из:

- 1) переработки и адаптации входящих данных;
- 2) тестирования новых данных;
- 3) развертывания обновленных компонентов системы;
- 4) минимального тестирования системы (так называемый `smoke-тест`).

⁶ https://github.com/aleksei-khitev/project_simulator/blob/master/report.py (дата обращения: 01.03.2025).

⁷ https://github.com/aleksei-khitev/project_simulator/blob/master/report.md (дата обращения: 01.03.2025).

⁸ <https://github.com/marktext/marktext> (дата обращения: 01.03.2025).

⁹ https://github.com/aleksei-khitev/project_simulator/blob/master/report.pdf (дата обращения: 01.03.2025).

¹⁰ https://github.com/aleksei-khitev/project_simulator/blob/master/project.py (дата обращения: 01.03.2025).

¹¹ https://github.com/aleksei-khitev/project_simulator/blob/master/project_iteration.py (дата обращения: 01.03.2025).

¹² https://github.com/aleksei-khitev/project_simulator/blob/master/team.py (дата обращения: 01.03.2025).

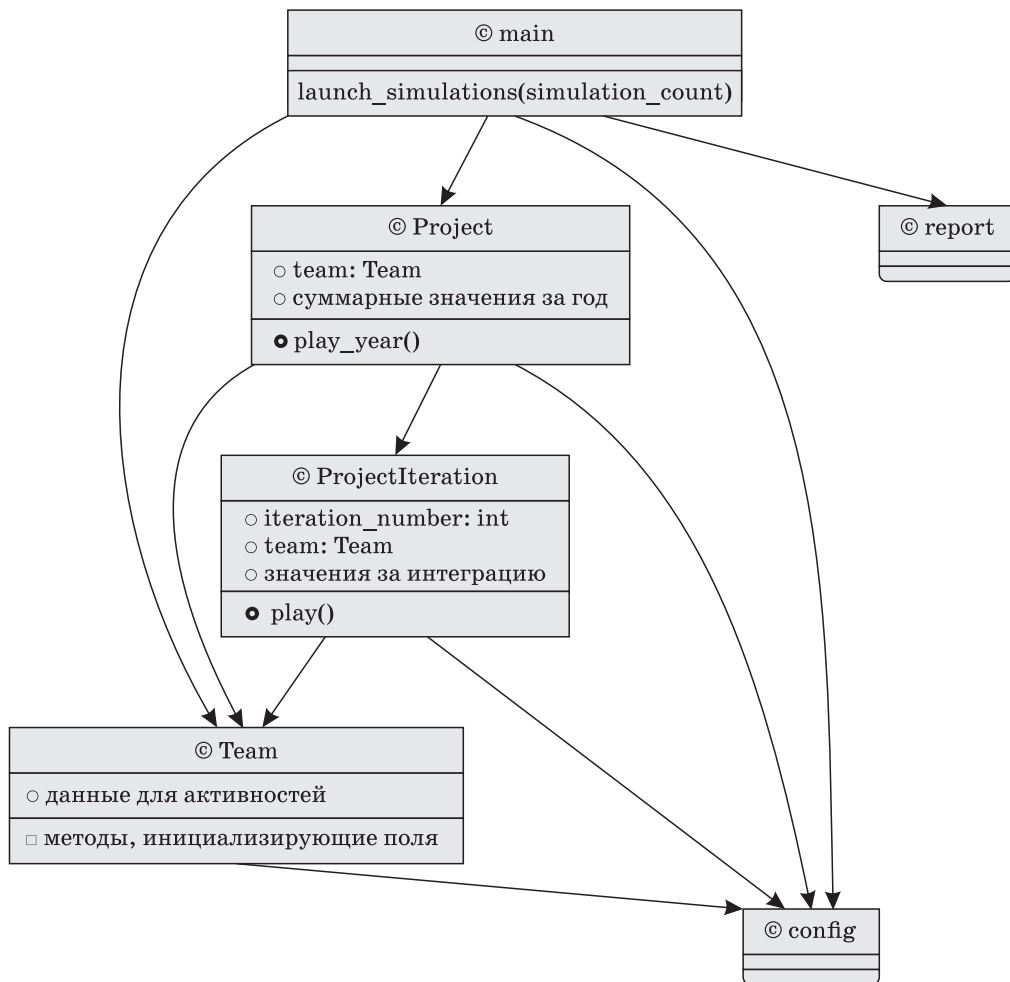


Рис 2. Упрощенная диаграмма классов и сценариев
 Fig. 2. Simplified Class and Scripts Diagram

Источник: [разработано авторами].

Все два дня релиза система была неспособна обслуживать запросы пользователей. Чтобы минимизировать вероятность увеличения простоя, в дни поставок новых версий выводились на работу дежурные:

- программный инженер;
- системный инженер;
- менеджер проекта;
- команда тестирования.

Непосредственно над проектом работала интернациональная команда из восьми человек. Подробно команда описана в табл. 2.

На проекте был налажен Scrum-подобный процесс, включающий ряд регулярных совещаний на каждую итерацию. Длительность итерации составляла 2 недели. Подробно совещания описаны в табл. 3.

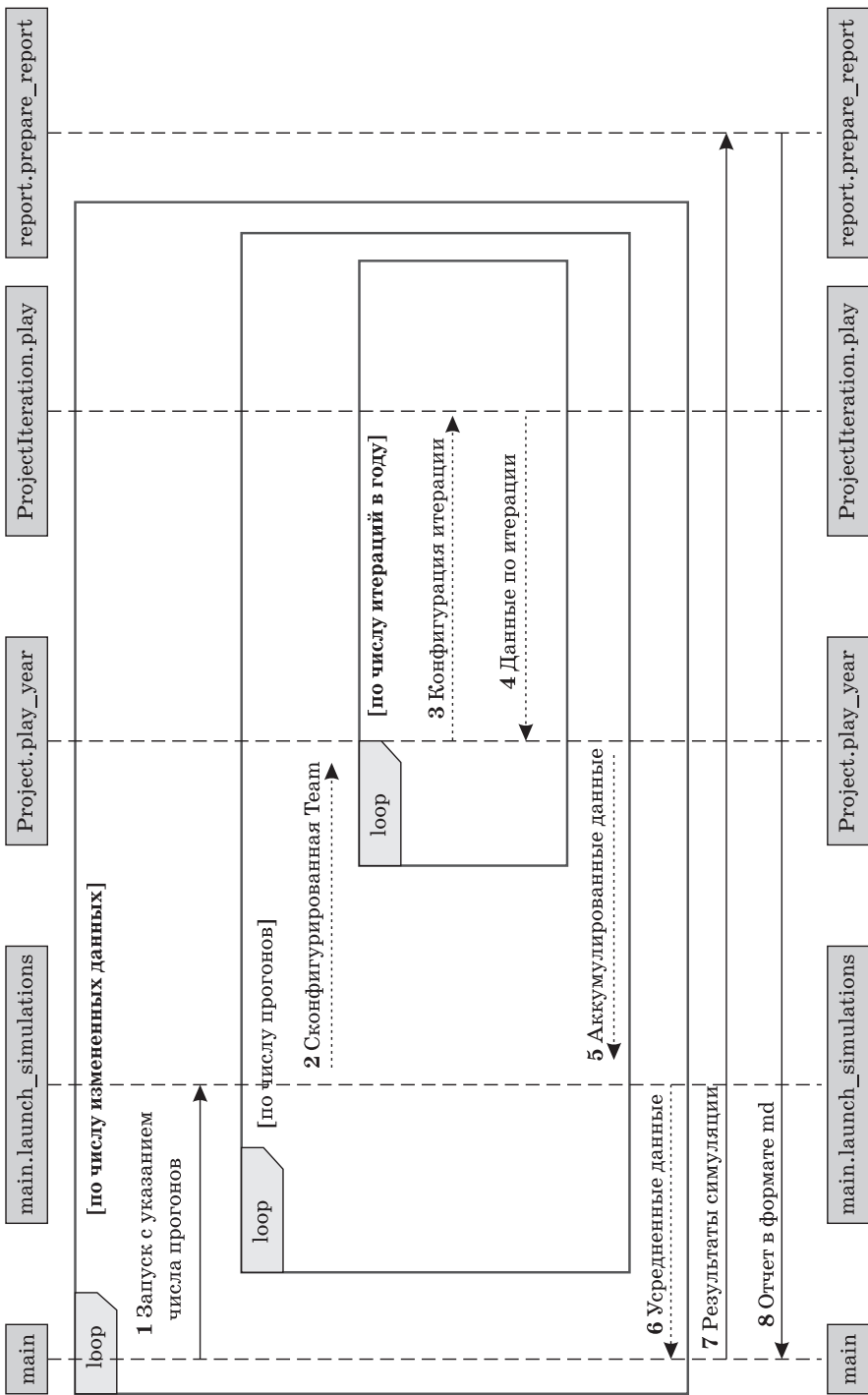


Рис 3. Диаграмма последовательности выполнения симуляции
 Fig. 3. Simulation Execution Sequence Diagram

Источник: [разработано авторами].

В ходе работы системы периодически возникали проблемы, которые требовалось решать в формате «тушения пожара». Можно выделить две часто встречающиеся проблемы, которые легко можно было бы решить, внедрив систему мониторинга. Проблемы описаны подробно в табл. 4.

Руководству было предложено внедрить систему мониторинга и предупреждения аварийных ситуаций (к примеру, zabbix), но положительного решения на установку дополнительного программного обеспечения добиться не удалось. Тогда в кратчайший срок была разработана собственная система мониторинга, включающая интерфейс пользователя и уведомления по электронной почте при приближении к пороговым значениям или выходе служб из строя. При этом данная система разворачивалась на уже доступном сервере приложений Tomcat, что не требовало установки дополнительного программного обеспечения.

Одной из организационных проблем в проекте была регулярная неподготовленность некоторых коллег к ежедневным совещаниям, усугубляющаяся невысоким владением английским языком. Это приводило к ежедневному перерасходу времени на утреннем совещании, включая драгоценное время проектного менеджера. Статусные совещания занимали от 25 минут вплоть до часа. Как показала практика в дальнейшем, время статусных совещаний может сократиться до 15 минут при том же составе команды.

Таблица 2

Состав команды описываемого проекта
Table 2. The team of the described project

Должность	Кол-во	Описание	Средняя зарплата в описываемый год (EUR)
Менеджер проекта	1	Великобритания	3240
Бизнес-аналитик	1	Испания	2640
Системный инженер	1	Испания	2280
Инженер по тестированию	1	Испания	2320
Программные инженеры	4	Россия, включая разработчиков на языках Perl, Java, JS и автоматизатора тестирования на Java	1443 (107 000 р. при курсе 74,12 рубля за 1 евро)

Таблица 3

Регулярные совещания проекта
Table 3. Regular project meetings

Название	Регулярность (на итерацию)	Средняя длительность	Вовлеченные члены команды
Статус	10	От 25 мин.	Все
Планирование итерации	1	1 ч.	Все
Приоритизация задач	1	1 ч.	Менеджер проекта, бизнес-аналитик
Оценка сроков выполнения задач	1	1 ч.	Все, кроме системного инженера
Демонстрация	1	1 ч.	Все
Ретроспектива	1	1 ч.	Все

Таблица 4

Регулярные аварийные ситуации

Table 4. Regular emergencies

Проблема	Среднее время поиска причины	Среднее время починки	Отвлеченные от работы сотрудники
Неинформативная ошибка из-за того, что в цепочке вызовов вышла из строя некая служба	15 мин.	10 мин.	Менеджер проекта, инженер по тестированию, системный инженер, разработчик
Неинформативная ошибка из-за того, что кончилось место на жестком диске на сервере	60 мин.	15 мин.	Менеджер проекта, инженер по тестированию, системный инженер, разработчик

Симуляция проекта с использованием программ для оптимизации

Указанные выше параметры проекта были заданы как параметры по умолчанию в конфигурации программы. Далее параметры менялись для различных случаев с учетом применения тех или иных решений.

Разработанная программа была запущена для четырех случаев, описанных в табл. 5.

Применение органайзера программного инженера позволяет в полуавтоматическом режиме подготовиться к совещаниям, а система заметок позволяет быстрее переключаться между различными задачами.

Система мониторинга позволяет принимать превентивные меры для исключения одной ситуации, мгновенного определения вышедшего сервиса во втором случае и понятных шагов исправления ситуации.

На рис. 4 показаны затраты времени на различные активности с применением улучшений и без них в течение года. Данные взяты из отчета, сгенерированного описанной программой.

На рис. 5 представлено аналогичное сравнение в разрезе затрат в течение года.

Таблица 5

Изменения в конфигурации для случаев

Table 5. Configuration changes for cases

Название	Изменения изначальной конфигурации
Изначальный	
С применением органайзера программного инженера [12, с. 37]	<ol style="list-style-type: none"> 1. Сокращается время ежедневных совещаний по статусу. 2. Изменяется распределение времени между разработкой нового функционала и исправлением дефектов с 40/50 на 60/30
С применением программы мониторинга [8, с. 265]	<ol style="list-style-type: none"> 1. Вероятность одного из представленных типов аварий снизилась до 0. 2. Время для расследования другого типа аварий снизилось до 0. 3. Участники, требуемые для исправления проблемы по второму типу аварий, сократились
Все вместе	Все вышеперечисленные

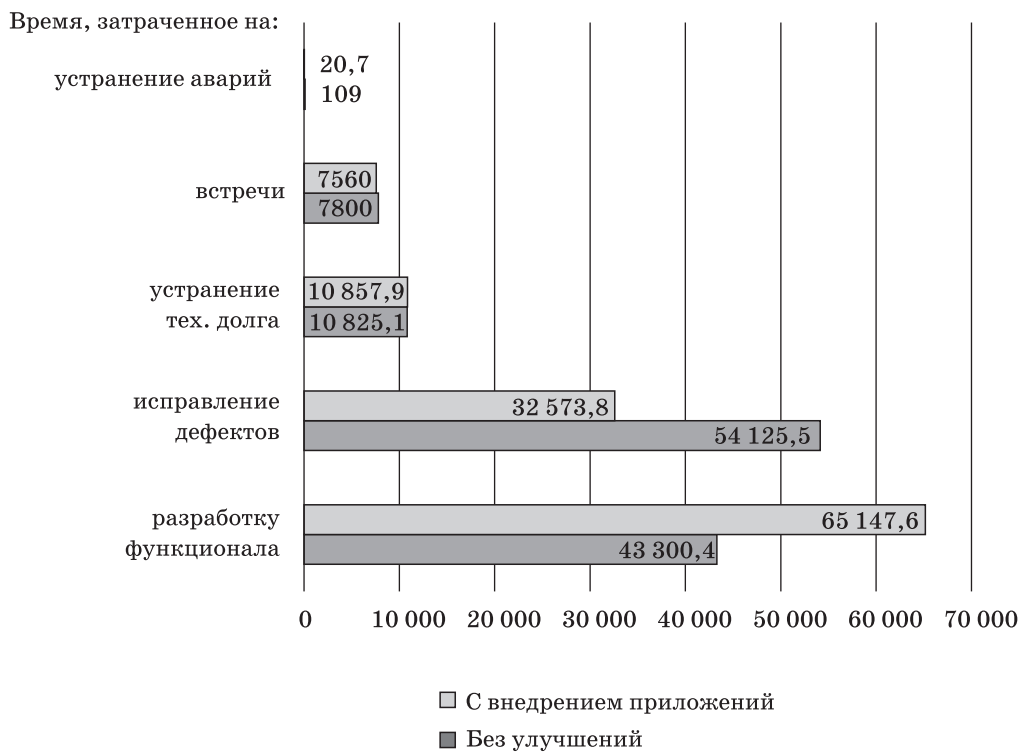


Рис. 4. Сравнение затрат времени
 Fig. 4. Time Cost Comparison

Источник: [разработано авторами].

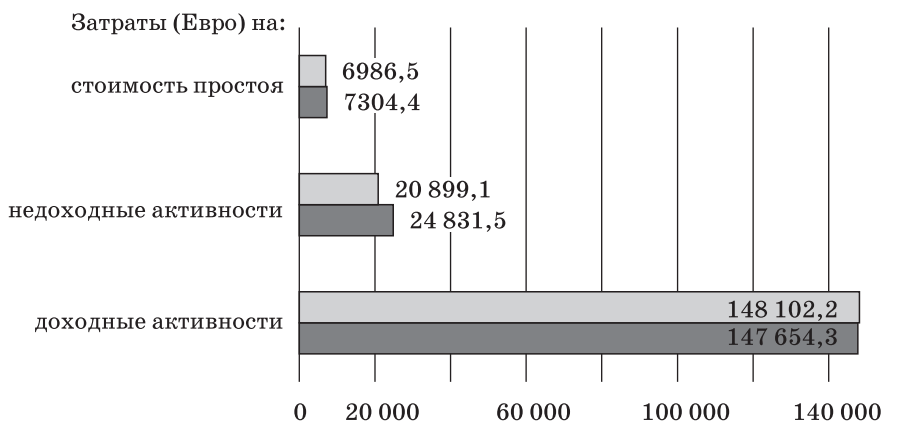


Рис. 5. Сравнение денежных затрат
 Fig. 5. Money Cost Comparison

Источник: [разработано авторами].

С экономической точки зрения, налицо снижение затрат на нерациональное расходование дорогостоящего времени специалистов, а также репутационные и иные потери из-за непредвиденных отказов в обслуживании системы.

С управленческой точки зрения, имеется более высокая предсказуемость работы команды и состояния проекта и оптимальное время утилизации ресурсов.

В дальнейшем и мониторинг и другие средства можно развивать, и тем самым повысить стабильность работы проекта. К примеру, в цикле статей компании ПАО Вымпел-Коммуникации (Билайн) описан путь развития мониторинга [3; 5; 4], а применив идеи из книги Максима Дорофеева [12], можно дополнительно усовершенствовать органайзер ИТ-специалиста.

На рис. 6 представлен фрагмент отчета о проведенном тесте.

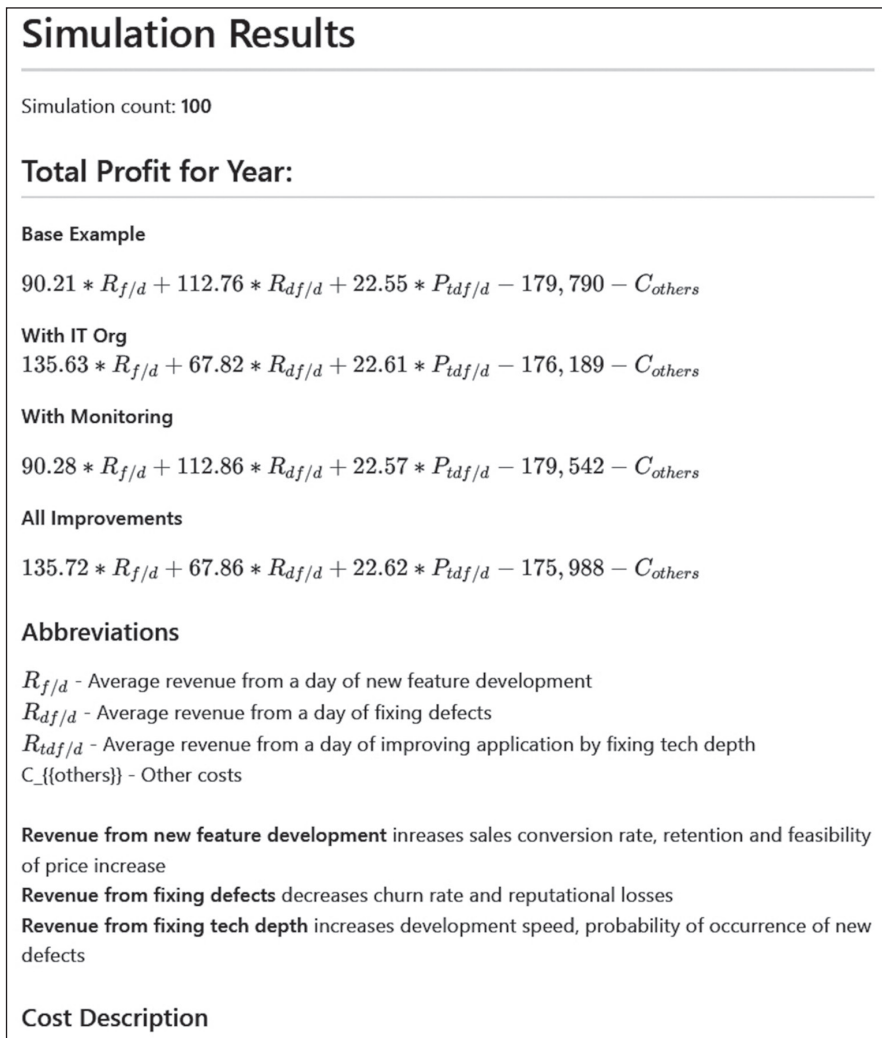


Рис. 6. Фрагмент отчета симуляции
 Fig. 6. Fragment of Simulation Report

Источник: [разработано авторами].

Результаты

С экономической точки зрения, налицо снижение затрат на нерациональное расходование дорогостоящего времени специалистов, а также репутационные и иные потери из-за непредвиденных отказов в обслуживании системы.

С управленческой точки зрения, имеется более высокая предсказуемость работы команды и состояния проекта и оптимальное время утилизации ресурсов.

Предложенную модель ИТ-проекта и разработанную на ее основе программ-симулятор выполнения проекта можно использовать для расчета эффекта от разнонаправленных оптимизаций с различной каденцией.

Заключение

В статье описана модель проекта, которая включает различные запланированные и незапланированные активности, затраты на выполнение и влияние указанных активностей на конечную прибыль от выполнения проекта.

Для симуляций с использованием предложенной модели разработана программа на языке программирования Python. С использованием программы рассмотрены несколько способов оптимизации работы проекта и рассчитан эффект от их внедрения.

Предложенная модель ИТ-проекта и разработанная на ее основе программа-симулятор выполнения проекта могут использоваться для расчета эффекта от разнонаправленных оптимизаций с различной каденцией. Модель можно усовершенствовать путем добавления в нее дополнительных активностей, например, процесса расширения команды и адаптации новых коллег.

Литература

1. Глущенко В. В. Проектная модель функционирования организации // The Scientific Heritage. 2020. N 53–3. P. 15–22.
2. Дорофеев М. Дждедайские техники. Как воспитать свою обезьяну, опустошить инбок и сберечь мыслетопливо. М.: Манн, Иванов и Фербер, 2020.
3. Как мы мониторим наши сервисы [Электронный ресурс]. URL: <https://habr.com/en/companies/beeline/articles/585536/> (дата обращения: 20.01.2025).
4. Как мы переводим наш мониторинг в наблюдаемость [Электронный ресурс]. URL: https://habr.com/en/companies/beeline_tech/articles/767018/ (дата обращения: 20.01.2025).
5. Как мы работаем с мониторингом и чем он нам помогает [Электронный ресурс]. URL: https://habr.com/en/companies/beeline_tech/articles/729504/ (дата обращения: 20.01.2025).
6. Михайлов Н. Э., Самойлова И. А., Горлачева Е. Н. Разработка формальной модели взаимодействия проектной команды // Инновации в менеджменте. 2021. № 1. С. 28–36.
7. Руководство к Своду знаний по управлению проектами (Руководство PMBOK®). М.: Издательство «Олимп-Бизнес», 2019.
8. Хитёв А., Юрков А. Система централизованного управления серверами на языке JAVA // Материалы III Международной научно-практической конференции «Устойчивое развитие: общество и экономика», 20–23 апреля 2016 г. СПб.: Скифия-принт, 2016. С. 265–266.
9. Abran A. Software project estimation: the fundamentals for providing high quality information to decision makers. John Wiley & Sons, 2015.
10. Boehm B. W. Software Engineering Economics // IEEE Transactions on Software Engineering. 1984. Jan. Vol. SE-10. N 1. P. 4–21. doi: 10.1109/TSE.1984.5010193
11. Holubyeva T. P. Structural-Logic Model of Time Forecast of Software Development Project Realization // Scientific Works of Vinnytsia National Technical University. 2008. N 3.
12. Khitev A., Yurkov A. A Software Engineer Organizer — New Special Solution // Journal of Applied Informatics. 2020. Vol. 15. N 1 (85). P. 37–46.

Об авторах:

Хитёв Алексей Юрьевич, ведущий разработчик, соискатель кандидатской степени, Санкт-Петербургский государственный университет (Санкт-Петербург, Российская Федерация); alexkhitev@gmail.com

Юрков Александр Васильевич, доктор физико-математических наук, профессор кафедры информационных систем Российского государственного педагогического университета им. А. И. Герцена (Санкт-Петербург, Российская Федерация); ayurkov@yandex.ru

References

1. Glushenko V. V. Project Model of Enterprise Functioning // The Scientific Heritage. 2020. N 53–3. P. 15–22. (In Russ.)
2. Dorofeev M. Jedi Techniques. How to Train Your Monkey, Empty Your Inbox, and Save Your Thought Fuel. Moscow: Publishing house Mann, Ivanov & Ferber, 2020. (In Russ.)
3. How we monitor our services. URL: <https://habr.com/en/companies/beeline/articles/585536/> (accessed: 20.01.2025). (In Russ.)
4. How we translate our monitoring into observability. URL: https://habr.com/en/companies/beeline_tech/articles/767018/ (accessed: 20.01.2025). (In Russ.)
5. How we work with monitoring and how it helps us. URL: https://habr.com/en/companies/beeline_tech/articles/729504/ (accessed: 20.01.2025). (In Russ.)
6. Mikhailov N. E., Samoilova I. A., Goralcheva E. N. The Elaboration of the Formal Communication Model in a Project Team // Innovations in Management [Innovatsii v menedzhmente]. 2021. N 1. P. 28–36. (In Russ.)
7. Project Management Body of Knowledge (PMBOK). Project Management Institute, 2017.
8. Khitev A., Yurkov A. Centralized Server Management System Based on Java Programming Language // Materials of the III International Scientific and Practical Conference “Sustainable Development: Society and Economy”, April 20–23, 2016. Saint Petersburg: Publishing house Skifiya-Print, 2016. P. 265–266. (In Russ.)
9. Abran A. Software project estimation: the fundamentals for providing high quality information to decision makers. John Wiley & Sons, 2015.
10. Boehm B.W. Software Engineering Economics // IEEE Transactions on Software Engineering. Jan. 1984. Vol. SE-10. N 1. P. 4–21. doi: 10.1109/TSE.1984.5010193
11. Holubyeva T.P. Structural-Logic Model of Time Forecast of Software Development Project Realization // Scientific Works of Vinnytsia National Technical University. 2008. N 3.
12. Khitev A., Yurkov A. A Software Engineer Organizer — New Special Solution // Journal of Applied Informatics. 2020. Vol. 15. N 1 (85). P. 37–46.

About the authors:

Aleksei Yu. Khitev, lead developer, PhD student, Saint Petersburg State University, Saint Petersburg, Russia; alexkhitev@gmail.com

Aleksandr V. Yurkov, Doctor of Science (Physical and Mathematical), Professor of the Information Systems Department of Herzen University, Saint Petersburg, Russia; ayurkov@yandex.ru