УДК 004.94+004.413
doi:10.18720/SPBPU/2/id24-551

Ракич Тина [1],
студент;
Вьюненко Людмила Федоровна [2],
доцент, канд. физ.-мат. наук, доцент

ПЕРСПЕКТИВА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В УПРАВЛЕНИИ КОНСТРУИРОВАНИЕМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

[1] Россия, Санкт-Петербург, Санкт-Петербургский государственный университет, rakic.tina@outlook.com;
[2] Россия, Санкт-Петербург, Санкт-Петербургский государственный университет промышленных технологий и дизайна, vyunenko@yandex.ru

Аннотация. Современные тренды ведут к повышению сложности создаваемого программного обеспечения, что обуславливает повышение внимания к фазе конструирования при его разработке. Сложность управления конструированием заключается в наличии многочисленных взаимосвязанных факторов, влияние которых традиционными методами управления отследить невозможно. Имитационное моделирование предоставляет возможность применит системный подход к конструированию, предоставляя при этом интерпретируемые результаты и возможность тестирования различных управленческих решений.

Ключевые слова: управление конструированием программного обеспечения, имитационное моделирование, управленческие решения.

Tina Rakic [1],
Student;
Lydmila F. Vyunenko [2],
Associate Professor, Candidate of Physical and Mathematical Sciences

THE PERSPECTIVE OF SIMULATION MODELING IN SOFTWARE CONSTRUCTION MANAGEMENT

[1] St. Petersburg University, St. Petersburg, Russia, rakic.tina@outlook.com;
[2] Saint Petersburg State University of Industrial Technologies and Design, St. Petersburg, Russia, vyunenko@yandex.ru

Abstract. Modern trends lead to an increase in the complexity of the software being created, which leads to increased attention to the construction phase during software development. The difficulty of software construction management lies in the presence of many interconnected factors, whose effects cannot be grasped by traditional management methods. Simulation modeling provides the opportunity to apply a system approach to software construction while providing interpretable results and the ability to test various management decisions.

Keywords: software construction management, simulation modeling, management decisions.

Введение

На протяжении всего существования отрасли разработки программного обеспечения (далее – ПО) управление проектами разработки новых программных средств, представляло собой задачу высокого уровня сложности. Несмотря на шаг к повышению эффективности проектов по разработке ПО, которые сделали гибкие («agile») методологии управления, проблемы, возникшие на ранних этапах развития данной отрасли, характерны и на сегодняшний день. Речь идет о проблемах перерасходования средств, несоблюдения сроков, низкого качества итогового продукта и как следствие, неудовлетворенности конечных пользователей. Учитывая стремительные скорости развития данной отрасли и ее интеграции в другие сферы, а также растущую комплексность создаваемого ПО [11], намечается необходимость в более эффективных инструментах управления.

Современные тренды акцент ставят на реализацию проектов с меньшими ресурсами и более короткими сроками реализации, при этом предполагающие более сложный исходный код с более простым пользовательским интерфейсом [7]. В осуществлении таких проектов, которые, как правило, мелкого и среднего масштаба, центральное место занимает неотъемлемый этап при разработке любого ПО – этап конструирования, включающий в себя этапы кодирования и отладки, а также верификации, модульного и интеграционного тестирования. На этап конструирования в малых и средних проектах уходит больше 50 % всех трудозатрат и также известно, что мелкие ошибки конструирования способны привести к масштабным убыткам компаний [2]. Естественно, что такой процесс нуждается в тщательном управлении. Механизмы, позволяющие правильно управлять конструированием, позволять повысить успешность реализации проектов по разработке ПО, а также финального продукта, так как качество итогового программного обеспечения непременно зависит от качества процесса его создания.

На исход фазы конструирования будет влиять множество факторов, большинство которых управленческой природы. Эффективное управление конструированием ПО предполагает рассмотрение данного процесса как системы, характеризующейся сложными взаимосвязями между ее элементами. Такая система не представима в виде ментальных моделей, а нуждается в более сложных инструментах. Как отмечал Кенетт Г. Купер, «Управление проектами постоянно терпит неудачи, иногда весьма впечатляющие, потому что менеджеры не могу увидеть, как их действия влияют на эффективность проекта» [9, с. 11]. Инструментом, позволяющим получить представление о системе в целом, является имитационное моделирование.

Инструментарий имитационного моделирования позволяет «заглядывать в будущее», предсказывая ход осуществления фазы конструирования, а также влияние тех или иных управленческих решений на ее исход.

Использование имитационного моделирования позволяет принимать действия по решению проблем до того, как эти проблемы наступят. Проекты по конструированию ПО характеризуются неопределенностью, динамическим поведением и наличием механизмов обратной связи, которые являются основными источниками сложностей при их реализации [7]. В этих условиях предиктивный механизм имитационной модели призван предоставить ответ на вопрос «что, если...?» и позволяет заблаговременно рассмотреть последствия принимаемых руководством управленческих решений.

Принимая во внимание уже имеющийся положительный опыт изучения управления конструированием ПО с помощью имитационной модели [3, 4, 5, 6, 10], можно утверждать, что имитационное моделирование претендует быть эффективным управленческим инструментом, позволяющим повысить успешность проектов. Целью настоящей работы является представление возможностей предложенной имитационной модели для управления конструированием ПО.

1. Основные характеристики процесса конструирования ПО

Управление конструированием ПО несущественно отличается в своем понятийном аппарате от управления другими комплексными проектами. Оцениваются такие величины как объем задач/работ, срок выполнения проекта и рабочая сила необходимая для выполнения проекта. Менеджмент оперирует понятиями поток задач/работ по проекту, продуктивность команды, разрыв между восприятием и реальностью, цикл мониторинга и контроля. В то же время отношения между этими объектами обладают рядом особенностей, характерных только для деятельности по конструированию ПО.

Объем работ по проекту образуется на основании ранее разработанных требований к ПО, которые в течение всего процесса конструирования меняются, дорабатываются и дополняются и, таким образом, постоянно меняют объем работ по проекту. Макконнелл С. отмечает: «Всем нам хотелось бы надеяться, что как только клиент утвердил требования, никаких изменений не произойдет. Однако чаще всего клиент не может точно сказать, что ему нужно, пока не будет написан некоторый код. Если вы планируете жестко следовать требованиям, на самом деле вы собираетесь не реагировать на потребности клиента» [2, с. 37–38]. Для эффективного управления конструированием непостоянный характер требований нельзя игнорировать, тем более что он стал основным фактором возникновения гибких методологии управления [12].

Программисты выполняют работу с определенной скоростью, на которую будет влиять численность команды, ее продуктивность, приближение срока конца проекта, а также фактическое время, отводимое для работы над проектом. Эти факторы носят динамический характер. Численность персонала меняется в соответствии с прогрессом работ над

проектом – если есть подозрения, что проект не будет выполнен вовремя, менеджмент нанимает дополнительный персонал, причем нужно понимать, что увеличение численности персонала не является прямо пропорциональным увеличению скорости выполнения работы. Первый это отметил Брукс: «добавление людей на заключительных стадиях процесса создания ПО продлевает срок завершения проекта» [8, с. 25]. Причиной является наличие в команде программистов коммуникационных потерь, которые занижают скорость выполнения работы.

Продуктивность работы, кроме способностей программистов, будет зависеть и от некоторых технических характеристик. Несмотря на то, что эти характеристики отличаются для различных проектов и организаций, в рамках одного проекта они остаются постоянными [3, 10].

По мере выполнения работ сопоставляются фактический и ожидаемый срок выполнения проекта, что приводит к влиянию на продуктивность фактора давления сроков. При его отсутствии программисты уделяют работе над проектом некоторую часть рабочего времени. Когда очевидно, что проект не будет вовремя выполнен, программисты начинают больше времени посвящать проекту и работают более усердно. Этот эффект появляется за счет положительного давления сроков [3]. При отрицательном давлении сроков ситуация ровно наоборот, и скорость выполнения работы замедляется [3].

В условиях положительного давления сроков будет расти и количество ошибок, генерируемых в процессе кодирования. Эти ошибки, хотя дешевле для исправления, чем ошибки допущенные в фазах формирования требований и проектирования ПО, своим существенным количеством могут оказать значительное влияние на исход финального продукта [2]. Конструирование ПО обязательно включает деятельность по их обнаружению и исправлению. Эта деятельность является частью любого управления проектами и в моделях изображается в виде цикла исправления ошибок [1, 9].

Имитационная модель, позволяющая эффективно управлять конструированием ПО, должна позволить отслеживать взаимосвязи между указанными характеристиками.

2. Построение имитационной модели

2.1. Обоснование выбора парадигмы построения модели

Конструирование ПО представляет собой циклический процесс, где каждая задача проходит несколько итераций [1]. Между элементами, участвующими в проекте по конструированию ПО, существует большое количество обратных связей, которые приводят к немедленной реакции системы, превращающей ее в динамическую: результаты принимаемых решений определяют проблемы, с которыми придется столкнуться в будущем, осознание которых изменяет оценку текущего состояния.

Все дополнительно усложняется необходимостью принятия управленческих решений при попытке удовлетворить большое число противоречивых требований, предъявляемых к проекту, и ожидания участников проекта. Невозможно одновременно сократить ресурсы и сроки выполнения проекта, оставляя при этом функциональность конечного продукта без изменений. Последствия принимаемых решений наступают не сразу, а с некоторой временной задержкой и часто по своим масштабам превышают масштабы события-инициатора. Для конструирования ПО, также, характерно отсутствие возможности объективно оценить прогресс по выполненной работе [3, 5, 6], что проявляется в высокой степени неопределенности, преследующей данные проекты.

Эти характеристики делают необходимым применение системного подхода с определенной степенью абстракции, который осуществим в рамках парадигмы системной динамики.

2.2. Обоснование выбора системы моделирования

Для реализации модели выбрана среда имитационного моделирования AnyLogic. Учитывая цель построения модели – создание эффективного механизма управления конструированием ПО, преимуществом данной системы является возможность наглядного графического представления модели вместе с дополнительными графическими и интерактивными элементами позволяющие наблюдать за параметрами модели и осуществить их изменение после ее запуска. Таким образом, расширяется количество управленческих сценариев, которые модель позволит протестировать. Данный интерактивный интерфейс модели позволяет использование модели лицам, не являющимся специалистами в области имитационного моделирования.

2.3. Реализация имитационной модели

Любой проект до своего начала проходит через стадию планирования, на которой утверждается объем выполняемой работы, срок выполнения, состав команды и прочие его характеристики. Эти характеристики будут являться входными параметрами модели. Построение имитационной модели в рамках парадигмы системной динамики предполагает представление моделируемой системы в виде диаграмм потоков и накопителей. В любой момент времени состояние моделируемой системы можно описать значениями, хранящимися в накопителях. Эти значения изменяются с течением времени согласно заданным в системе потокам. Математически это означает, что модель можно формализовать в виде задачи Коши для системы обыкновенных дифференциальных уравнений. Реализация модели в виде диаграммы потоков и накопителей представлена на рис. 1.

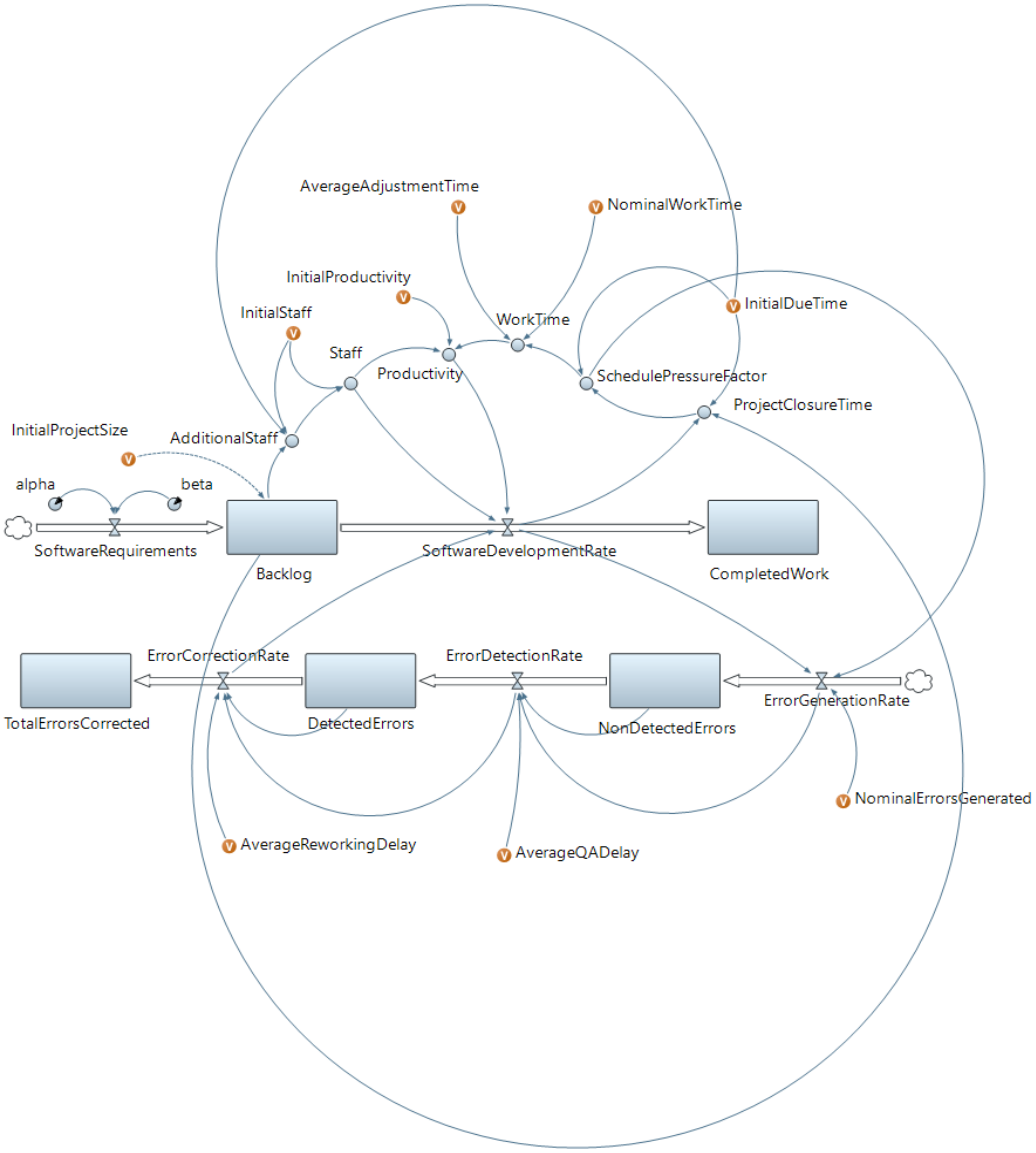

Рис. 1. Диаграмма потоков и накопителей процесса конструирования ПО

Запуск модели, представленный на рис. 2, дает возможность проследить динамику осуществления проекта. Таким образом, руководство сможет протестировать несколько сценариев осуществления проекта и выбрать наиболее подходящий вариант. Данная модель позволит получить ответы на вопросы о размере команды, необходимой для своевременного выполнение проекта, а также определить, сможет ли более мелкая команда справиться с необходимой работой. На этапе планирования сроков осуществления проекта модель способна ответить на вопрос, насколько вероятно его выполнение в заданный срок, а также насколько может быть обоснованно его сокращение. Однако самым полезным с управленческой точки зрения является возможность тестирования сразу комбинации факторов и получения ответа на вопросы следующего характера: если сократить срок выполнения на *x* дней и включить в проект *y* новых сотрудников, можно ли добиться его выполнения в срок и хорошего качества создаваемого ПО.

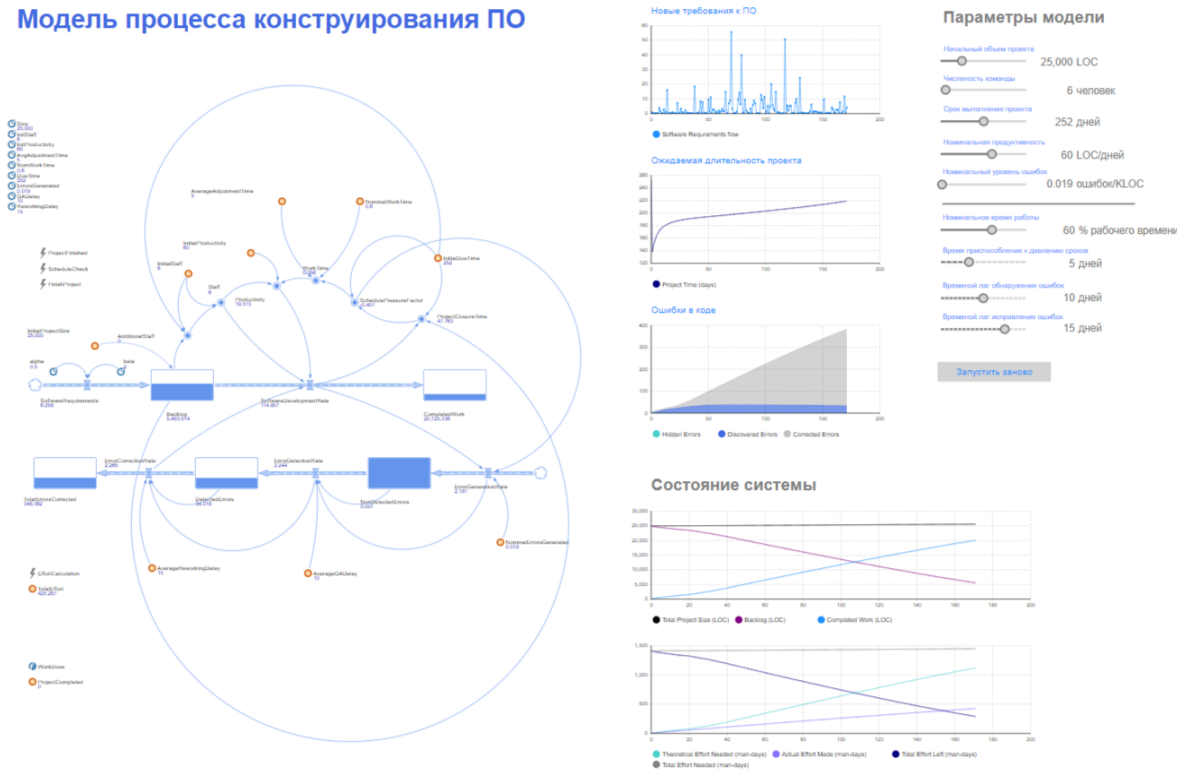

Рис. 2. Модель процесса конструирования ПО в среде AnyLogic

Заключение

В работе обсуждается вопрос повышения качества управления фазой конструирования ПО с использованием инструментария имитационного моделирования. Выявлены основные объекты, участвующие в конструировании ПО, и дана характеристика их взаимосвязей. Предложена архитектура системно-динамической имитационной модели, которая бы позволила руководству проекта протестировать различные управленческие политики до начала проекта и определить оптимальную стратегию действий, повышая таким образом, эффективность принимаемых управленческих решений.

Список литературы

1. Каталевский Д. Ю., Суслов С. А. Имитационное моделирование в управлении сложными проектами // Проблемы теории и практики управления. – 2022. – Т. 2. – № 2. – С. 101–116.

2. Макконнелл С. Совершенный код: практическое руководство по разработке программного обеспечения. Мастер-класс / Пер. с англ. – М.: Издательство «Русская редакция», 2010. – 896 с.

3. Abdel-Hamid T. K. The dynamics of software development project management: An integrative system dynamics perspective: PhD thesis. – Massachusetts: Massachusetts Institute of Technology, 1984.

4. Abdel-Hamid T. K., Madnick S. E. Impact of schedule estimation on software project behavior // IEEE Software – 1986. – Vol. 3 – Pp. 70–75. – DOI: 10.1109/MS.1986.234072.

5. Abdel-Hamid T. K., Madnick S. E. Lessons learned from modeling the dynamics of software development // Communications of the ACM 32 – 1989. – Vol. 12 – Pp. 1426–1438.

6. Abdel-Hamid T. K., Madnick S. E. Modeling the dynamics of software project management. – Massachusetts: Massachusetts Institute of Technology, 1987.

7. New trends in software process modeling / S. T. Acuna, M. I. Sanchez-Segura (eds.) // Series on software engineering and knowledge engineering. – World Scientific Publishing Co., 2006. – Vol. 18.

8. Brooks F. P. Jr. The mythical man-month: Essays on software-engineering – Addison Wesley Publishing Company, 1975.

9. Cooper K. G. The $2,000 hour: How managers influence project performance trough rework cycle // Project Management Journal – 1994. – Vol. 25 – Pp. 11–23.

10. Lin Chi Y., Abdel-Hamid T. K., Sherif J. S. Software-engineering process simulation model (SEPS) // Journal of Systems and Software – 1997. – Vol. 38(3) – Pp. 263 – 277.

11. Strålin T., Gnanasambandam Ch., Andén P., Comella-Dorda S., Burkacky O. Software development handbook. Transforming for the digital age [Electronic resource] / McKinesy & Company, Inc., 1996–2024. – URL: https://www.mckinsey.com/~/media/McKinsey/Industries/Technology%20Media%20and%20Telecommunications/High%20Tech/Our%20Insights/Software%20Development%20Handbook%20Transforming%20for%20the%20digital%20age/Software%20Development%20Handbook%20Transforming%20for%20the%20digital%20age.pdf (access date: 22.05.2024).

12. Wiegers K. E., Beatty J. Software requirements. – Washington, DC: Microsoft Press, 2013.