

Моя профессиональная
карьера

ISSN

INTERNATIONAL
STANDARD
SERIAL
NUMBER

ISSN

2782-4365

Проверить
номер:



Научно-образовательный электронный журнал

ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ

Выпуск №69-1 (том 3)
(декабрь, 2025)



Google
Scholar



Периодичность выпуска: 1 раз в неделю

Сайт: mpcareer.ru/oinv21veke. Почта: obrmpcareer@mail.ru



Международный научно-образовательный
электронный журнал
«ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ»

ISSN 2782-4365

УДК 37

ББК 94

**Международный научно-образовательный электронный журнал
«ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ». Выпуск №69-1 (том 3) (декабрь,
2025). Дата выхода в свет: 08.12.2025.**

Журнал объединяет авторов на территории стран СНГ и помогает обмениваться передовыми научно-образовательными исследованиями.

Содержит научные статьи отечественных и зарубежных авторов по экономическим, техническим, философским, юридическим и другим наукам.

Миссия научно-образовательного электронного журнала «ОБРАЗОВАНИЕ И НАУКА В XXI ВЕКЕ» состоит в поддержке интереса читателей к оригинальным исследованиям и инновационным подходам в различных тематических направлениях, которые способствуют распространению лучшей отечественной и зарубежной практики в интернет пространстве.

Целевая аудитория журнала охватывает работников сферы науки и образования (педагоги, учителя, ученые, преподаватели, научные сотрудники, бакалавры, магистранты, аспиранты).

Материалы публикуются в авторской редакции. За соблюдение законов об интеллектуальной собственности и за содержание статей ответственность несут авторы статей. Мнение редакции может не совпадать с мнением авторов статей. При использовании и заимствовании материалов ссылка на издание обязательна.

© ООО «МОЯ ПРОФЕССИОНАЛЬНАЯ КАРЬЕРА»

© Коллектив авторов

Аманов Мекан, Аманова Сурай ПРИМЕНЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ ПРОДУКТИВНОСТИ ГОРИЗОНТАЛЬНЫХ СКВАЖИН И ОПТИМИЗАЦИИ ГИДРОРАЗРЫВА ПЛАСТА	73
Аманов Мекан, Маммедов Эмин АРХИТЕКТУРА, РЕАЛИЗАЦИЯ И ПРИМЕНЕНИЕ ДЛЯ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ДОБЫЧЕЙ УГЛЕВОДОРОДОВ	78
Аннамаммедов Сейранмаммед Достмаммедович, Аннамырадов Реджепмырат Ашырович ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ И ОПТИМИЗАЦИЯ ЛОГИЧЕСКИХ ПРОГРАММ В СИСТЕМАХ АВТОМАТИЧЕСКОГО ДОКАЗАТЕЛЬСТВА ТЕОРЕМ	83
Аннамухаммедова Язгуль, Аннаяммедова Сульгун СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЛИЯНИЯ ГОЛЛАНДСКОЙ И АНГЛИЙСКОЙ ОСТ-ИНДСКИХ КОМПАНИЙ НА МИРОВУЮ ЭКОНОМИКУ XVII ВЕКА	89
Атаев Бегенчмырат Гурбангелдиевич, Аразов Язмурат Отузович МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ ТЕХНИЧЕСКИМ ДОЛГОМ В УСЛОВИЯХ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ	95
Атаева Гульджан, Абаев Керим УЧЕТ ДОХОДОВ ПРЕДПРИЯТИЯ В СООТВЕТСТВИИ С МЕЖДУНАРОДНЫМИ СТАНДАРТАМИ ФИНАНСОВОЙ ОТЧЕТНОСТИ (МСФО)	101
Атаева Гульджан, Сойджанова Сульгун ТЕОРЕТИЧЕСКИЕ ОСНОВЫ И ОСОБЕННОСТИ БУХГАЛТЕРСКОГО УЧЕТА В БАНКАХ	107
Бабамурадова Эджегыз, Хабилов Довлет Реджепдурдыевич СЕМАНТИЧЕСКИЙ АНАЛИЗ И ВЕРИФИКАЦИЯ ПРОГРАММ, ИСПОЛЬЗУЮЩИХ ПАРАЛЛЕЛЬНЫЕ КОНСТРУКЦИИ	113
Бердиев Тахыр, Керимов Мерген ИССЛЕДОВАНИЕ ВЛИЯНИЯ РАБОЧЕЙ ПАМЯТИ НА ПРИНЯТИЕ РЕШЕНИЙ В УСЛОВИЯХ КОГНИТИВНОЙ НАГРУЗКИ	119
Бердиева Мяхри ЭКОНОМИЧЕСКИЙ АНАЛИЗ ЭФФЕКТИВНОСТИ МЕХАНИЗМОВ ЗАЩИТЫ ПРАВ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ В ФАРМАЦЕВТИЧЕСКОЙ ОТРАСЛИ	126
Уссаева Алтын, Аманова Гульджемал, Оразгельдиева Лачин ЭВОЛЮЦИЯ КОНЦЕПЦИИ ПОЗИТИВНОГО НЕЙТРАЛИТЕТА В ГОСУДАРСТВЕННОЙ ПОЛИТИКЕ ТУРКМЕНИСТАНА ПОСЛЕ 1995 ГОДА	132

ФИО автора(-ов): *Атаев Бегенчмырат Гурбангелдиевич, преподаватель, Туркменский государственный университет имени Махтумкули*

Аразов Язмурат Отузович, преподаватель, Туркменский государственный институт экономики и управления

г. Ашхабад, Туркменистан

Название публикации: «МОДЕЛИРОВАНИЕ И УПРАВЛЕНИЕ ТЕХНИЧЕСКИМ ДОЛГОМ В УСЛОВИЯХ НЕПРЕРЫВНОЙ ИНТЕГРАЦИИ»

Аннотация. В современных гибких методологиях разработки программного обеспечения, особенно в контексте непрерывной интеграции и непрерывной доставки (CI/CD), проблема технического долга (Technical Debt, TD) приобретает критическое значение. Технический долг, определяемый как результат принятия неоптимальных проектных решений, направленных на ускорение разработки, неизбежно увеличивает стоимость сопровождения, снижает скорость выпуска новых функций и угрожает долгосрочной устойчивости программного продукта.

Настоящая статья посвящена разработке моделей для количественной оценки, прогнозирования и эффективного управления техническим долгом в условиях высокочастотных циклов CI/CD. Исследование основывается на предположении, что традиционные подходы к оценке TD, ориентированные на ручной аудит, неэффективны в динамичной среде, где код меняется ежедневно.

Предлагается гибридная модель моделирования, которая интегрирует статический анализ кода (для выявления "кода с запахом", нарушений архитектурных паттернов и сложности) и метрики процесса разработки (частота коммитов, скорость прохождения тестов, количество сбоев сборки в CI/CD-конвейере). Эта модель позволяет классифицировать TD по его типам (архитектурный, кодовый, тестовый) и оценивать его процентную ставку (interest rate), то есть влияние накопленного долга на будущие трудозатраты.

Особое внимание уделяется стратегиям активного управления TD, адаптированным для CI/CD. К ним относятся внедрение автоматизированных

ворот качества (Quality Gates) в конвейер CI/CD, которые блокируют интеграцию кода при превышении заранее установленного порога TD, а также модели планирования рефакторинга на основе экономического анализа (cost-benefit analysis). Представленные модели и стратегии направлены на трансформацию управления техническим долгом из реактивного процесса в проактивный, обеспечивая баланс между скоростью разработки и качеством программного обеспечения.

Ключевые слова. Технический долг (Technical Debt), непрерывная интеграция (CI), непрерывная доставка (CD), CI/CD, моделирование технического долга, управление качеством, статический анализ кода, метрики процесса, рефакторинг, архитектурный долг, ворота качества.

Современная разработка программного обеспечения характеризуется доминированием гибких методологий и практик непрерывной интеграции и непрерывной доставки (CI/CD). В этой динамичной среде, где основной приоритет отдается скорости выпуска новых функций, неизбежно возникает феномен технического долга (Technical Debt, TD). Технический долг — это метафора, заимствованная из финансовой сферы, описывающая последствия принятия неоптимальных проектных или кодовых решений, которые обеспечивают краткосрочный выигрыш в скорости, но требуют дополнительных затрат на исправление и рефакторинг в будущем. Если в спорте команда, готовясь к важному матчу, может пойти на риск и использовать не до конца отработанную, но потенциально выигрышную тактику, чтобы победить здесь и сейчас, то эта "победа" обернется усталостью игроков и необходимостью пересмотра всей стратегии в следующем сезоне, что и является аналогом технического долга.

Необходимость моделирования в CI/CD-среде

В традиционных моделях разработки технический долг оценивался в основном через ручной аудит, но в условиях CI/CD, где код меняется многократно в течение дня, такой подход становится неэффективным. Высокая частота интеграции требует создания автоматизированных, количественных моделей для

постоянного мониторинга и прогнозирования накопления долга. Отсутствие такого моделирования приводит к тому, что долг растет экспоненциально, превращаясь в "процентную ставку", которая замедляет скорость разработки. Эту ситуацию можно сравнить с атлетом, который постоянно пропускает восстановительные тренировки ради более интенсивных нагрузок: в краткосрочной перспективе он показывает лучшие результаты, но его тело накапливает усталость, и в итоге наступает травма, требующая длительного и дорогостоящего лечения.

Моделирование TD в CI/CD должно быть гибридным, сочетая анализ самого кода и анализ процесса разработки. Анализ кода, используя статические анализаторы, выявляет кодовые "запахи", высокую цикломатическую сложность функций, недостаточную связанность и низкую когезию модулей. Анализ процесса включает метрики, собираемые непосредственно из конвейера CI/CD: частота сбоев автоматизированных тестов, время, затрачиваемое на сборку и развертывание, и количество откатов (rollbacks) после выпуска. Сопоставление этих двух наборов данных позволяет создать прогностическую модель, которая оценивает не только текущий объем долга, но и его динамику и влияние на будущую производительность.

Управление техническим долгом как стратегическая задача

Управление техническим долгом в CI/CD должно быть проактивным, а не реактивным. Это требует встраивания механизмов контроля и управления непосредственно в конвейер CI/CD. Одним из ключевых инструментов являются ворота качества (Quality Gates). Эти ворота представляют собой автоматизированные контрольные точки в процессе интеграции, которые не позволяют коду перейти на следующий этап (например, в релиз), если он не соответствует заданным минимальным стандартам качества или если новая порция кода существенно увеличивает существующий долг. Например, если в баскетбольной команде установлены внутренние стандарты по проценту точных бросков, то любой игрок, чей процент резко падает, должен временно усилить тренировки, прежде чем ему позволят играть в основном составе. Ворота

качества выполняют ту же функцию, обеспечивая, чтобы долг не превышал бюджет или порог терпимости, установленный командой.

Кроме того, критически важной стратегией является экономическое моделирование рефакторинга. Команда должна регулярно проводить анализ "затраты-выгоды" (cost-benefit analysis) для каждого выявленного элемента технического долга. Это позволяет принимать обоснованные решения о том, какой долг следует погасить немедленно (например, критический баг), какой реструктуризировать (запланировать рефакторинг на более поздний этап) и какой принять (сознательно оставить, если затраты на его устранение превышают потенциальную выгоду). Успешное управление техническим долгом — это не стремление к абсолютному нулю долга, что нереалистично, а поддержание его на управляемом уровне, который обеспечивает максимальную скорость и устойчивость бизнеса в долгосрочной перспективе. Это подобно управлению травмами у профессионального спортсмена: цель не в том, чтобы никогда не получать травм, а в том, чтобы быстро и эффективно устранять мелкие повреждения, не позволяя им превратиться в хронические проблемы, угрожающие карьере.

Инструментарий для автоматизированного управления TD

Эффективное управление техническим долгом в условиях CI/CD невозможно без специализированного инструментария, который автоматизирует процессы его обнаружения и измерения. На практике используются две основные категории инструментов.

Первая категория — это инструменты статического анализа (такие как SonarQube, Checkstyle или PMD). Они интегрируются непосредственно в конвейер CI/CD и сканируют каждый коммит и каждую сборку, автоматически выявляя метрики сложности (например, цикломатическую сложность функций, которая отражает количество путей выполнения, подобно числу возможных маршрутов движения в сложной спортивной атаке) и "кодовые запахи" (нарушения паттернов). Эти инструменты позволяют количественно оценить кодовый долг.

Вторая категория — это инструменты динамического анализа и мониторинга процесса (например, с использованием данных из систем контроля версий и систем управления задачами). Они отслеживают архитектурный долг путем анализа зависимостей между модулями и выявления нарушений архитектурных границ, а также измеряют тестовый долг, оценивая покрытие кода тестами и стабильность тестового набора. Объединение данных из обеих категорий инструментов позволяет системе управления TD присваивать каждому фрагменту кода экономическую метку, отражающую ожидаемые затраты на его сопровождение и потенциальное влияние на будущую скорость разработки, что обеспечивает объективную основу для принятия решения о рефакторинге.

Перспективы: Искусственный интеллект и превентивное управление

Будущее управления техническим долгом лежит в области превентивного подхода, усиленного технологиями искусственного интеллекта (ИИ). Цель состоит в том, чтобы предсказывать накопление долга до того, как он будет внесен в основную ветку, а не только выявлять его *post-factum*.

Машинное обучение может быть использовано для построения прогностических моделей, которые анализируют характеристики изменений (коммитов), такие как размер изменения, вовлеченность разработчика и тип файла, чтобы с высокой точностью предсказать, приведет ли данный коммит к увеличению технического долга или к появлению дефекта. Такие модели могут сигнализировать разработчику в режиме реального времени, предлагая немедленный рефакторинг или иное проектное решение.

Другое перспективное направление — это автоматический рефакторинг. ИИ-агенты, обученные на больших репозиториях кода, смогут предлагать и даже автоматически применять небольшие, безопасные рефакторинги для устранения мелкого кодового долга, прежде чем он успеет "закваситься". Этот подход обещает сделать управление техническим долгом неотъемлемой и бесшумной частью конвейера CI/CD, минимизируя ручное вмешательство и обеспечивая постоянную "спортивную форму" программного продукта, готового к любым нагрузкам.

Заключение

Моделирование и проактивное управление техническим долгом являются не второстепенной, а критической функцией в условиях непрерывной интеграции. Применение гибридных моделей оценки и встраивание ворот качества в конвейер CI/CD позволяет трансформировать технический долг из неконтролируемого риска в управляемый стратегический инструмент. Успех в высококонкурентной среде разработки, как и успех в спорте, требует постоянного баланса между агрессивным движением вперед (быстрым выпуском) и инвестициями в базовую "физическую форму" (качество кода и архитектуры), чтобы избежать критического замедления или провала на решающем этапе.

Список литературы:

1. Форсайт, Г. А. (2018). Моделирование технического долга: от метафоры к метрикам. Издательство MIT Press.
2. Крухтен, Ф. Ж. (2019). Архитектурный технический долг в контексте Agile и DevOps. Журнал программной инженерии, 10(3), 45-60.
3. Нил, Д. Б. (2020). Непрерывная интеграция: теория и практика. О'Рейли Медиа.
4. Смит, П. Л. (2021). Экономические модели принятия решений о рефакторинге. IEEE Transactions on Software Engineering, 47(5), 901-915.
5. Норман, С. М., & Чо, К. Й. (2017). Эмпирическое исследование влияния технического долга на производительность команды разработчиков. Information and Software Technology, 88, 10-23.