

УДК 004.4; 005.1

DOI: [10.26102/2310-6018/2026.56.5.004](https://doi.org/10.26102/2310-6018/2026.56.5.004)

Особенности оценки эффективности процессов в ИТ-команде при применении гибких методологий управления

М.Р. Котова✉

Инвестиционный банк Барклайс, Прага, Чешская Республика

Резюме. Оценка эффективности процессов в ИТ-командах, которые применяют гибкие методологии управления (Agile), представляет собой актуальную задачу, связанную с необходимостью сокращения времени поставки изменений при одновременном обеспечении устойчивости (качества) и экономической целесообразности разработки. Традиционные результатные показатели, ориентированные на сроки и объем выполненных работ, оказываются недостаточно информативными в условиях итеративной и инкрементной разработки, поскольку не отражают вариативность потока, потери и последствия снижения качества. В статье предложена процессно-ориентированная модель оценки эффективности ИТ-команды, основанная на агрегировании метрик потока работ, устойчивости поставки и экономических потерь в интегральный индекс процессной эффективности. Модель опирается на данные цифровых следов жизненного цикла ИТ-продукта, которые формируются в трекарах задач, системах управления версиями, CI/CD-пайплайнах и средствах мониторинга. В рамках исследования обосновано использование мультипликативной формы агрегирования, что позволяет учитывать влияние лимитирующих факторов процесса. Апробация модели на данных продуктовых команд, которые применяют Agile-подходы, подтвердила, что интегральная оценка позволяет выявлять деградацию процессов на ранних этапах и локализовать проблемные зоны, связанные с управлением потоком, устойчивостью поставки и накоплением технического долга. Полученные результаты демонстрируют возможность использования предложенной модели в качестве инструмента поддержки управленческих решений и регулярного мониторинга эффективности ИТ-команд.

Ключевые слова: разработка по принципам Agile, Lean, DevOps, эффективность процессов, метрики потока, устойчивость поставки, технический долг, интегральный индекс, ИТ-команда.

Для цитирования: Котова М.Р. Особенности оценки эффективности процессов в ИТ-команде при применении гибких методологий управления. *Моделирование, оптимизация и информационные технологии*. 2026;14(5). URL: <https://moitvvt.ru/ru/journal/article?id=2221> DOI: 10.26102/2310-6018/2026.56.5.004

Features of evaluating process efficiency in an IT team when applying Agile management methodologies

M.R. Kotova✉

Barclays Investment Bank, Prague, Czech Republic

Abstract. Evaluating the efficiency of processes in IT teams applying agile management methodologies (Agile) is a relevant research problem associated with the need to reduce change delivery lead time while simultaneously ensuring delivery stability (quality) and the economic feasibility of software development. Traditional outcome-based indicators focused on deadlines and the volume of completed work prove to be insufficiently informative in iterative and incremental development contexts, as they fail to reflect flow variability, process losses, and the consequences of quality degradation. This paper proposes a process-oriented model for evaluating IT team efficiency, based on aggregating flow metrics, delivery stability metrics, and economic loss indicators into an integral process efficiency index. The model relies on digital trace data generated throughout the IT product life cycle in issue tracking

systems, version control systems, CI/CD pipelines, and monitoring tools. Within the framework of the study, the use of a multiplicative aggregation approach is substantiated, which makes it possible to account for the impact of limiting factors in the development process. The approbation of the model using data from product teams applying Agile approaches confirms that the integral assessment enables early detection of process degradation and the localization of problem areas related to flow management, delivery stability, and the accumulation of technical debt. The results obtained demonstrate the feasibility of using the proposed model as a tool for managerial decision support and continuous monitoring of IT team process efficiency.

Keywords: Agile development, Lean, DevOps, process efficiency, flow metrics, delivery stability, technical debt, integral index, IT team.

For citation: Kotova M.R. Features of evaluating process efficiency in an IT team when applying Agile management methodologies. *Modeling, Optimization and Information Technology*. 2026;14(5). (In Russ.). URL: <https://moitvvt.ru/ru/journal/article?id=2221> DOI: 10.26102/2310-6018/2026.56.5.004

Введение

Активное распространение гибких методологий управления в командах разработчиков информационных технологий (ИТ) обусловлено потребностью организаций сокращать время вывода изменений на рынок и одновременно повышать способность продуктов эволюционировать под воздействием факторов внешней неопределенности. В этой связи основное внимание смещается от подходов, основанных на разовой сдаче проекта, к непрерывной поставке изменений для поддержания работоспособности и развития цифровых сервисов и продуктов. Однако существующие эмпирические исследования Agile указывают на неоднородность и нередкую противоречивость результатов внедрения гибкого управления, поскольку эффекты зависят от множества сопутствующих факторов (среди которых, например, зрелость и компетенции инженеров, структура зависимостей, качество входящего потока требований, регламентов принятия решений, наличие организационных ограничений и т. п.) [1]. Следовательно, задача корректной оценки эффективности процессов в ИТ-команде при применении гибких методологий управления становится проблемной. При отсутствии валидной модели измерений эффективности нарушается способность команды выявлять проблемы и эффективно управлять улучшениями продукта.

Отдельно стоит указать, что традиционная оценка эффективности ИТ-процессов зачастую предполагает использование статических показателей (учитываются сроки, бюджет и объем) и отчетности по выполнению планов. Ограничением данного подхода является фиксированность входа и линейность перехода от ресурсов к результату, что делает его попросту неприменимым в условиях разработки по итерациям, поскольку: объем работ уточняется в ходе разработки; ценность изменений формируется через обратную связь и последовательные инкременты; существенная часть потерь возникает не в «выполнении», а в ожидании, изменениях, при накоплении технического долга и снижении качества решения. Поэтому при использовании традиционных оценок их эффективность и интерпретируемость снижается.

Дополнительным и не менее важным фактором видится принятие на практике попыток механически перенести в Agile метрики производительности (например, скорость в сторипointах (Story Points) на уровень сравнения ИТ-команд и контроля их эффективности. В конечном счете возникают риски подмены качества количеством. В современных исследованиях, посвященных улучшению процессов в agile, указывается необходимость использования метрик процесса с инструментами software process improvement (SPI), которые должны соответствовать итеративному развитию и непрерывному улучшению [2]. Отсюда следует главное противоречие, выраженное в

том, что оценка эффективности в Agile должна быть процессной, опираться на цифровые следы жизненного цикла и включать в себя элементы измерения качества, надежности (устойчивости ИТ-продукта) и потерь.

В совокупности представленные обстоятельства определили цель и границы настоящего исследования.

Цель исследования – разработать модель оценки эффективности процессов ИТ-команды при применении гибких методологий управления, которая будет обеспечивать: сопоставимость показателей во времени и в пределах схожего класса команд; возможность сравнения и интерпретации при управлении процессами в ИТ-команде; связанность с практиками улучшения потока, качества и экономической эффективности ИТ-продуктов.

Исходя из заявленной цели исследования формулируются следующие его задачи:

- 1) уточнить теоретические основания категории «эффективность процессов» применительно к разработке ИТ-продуктов и деятельности ИТ-команды;
- 2) описать трансформацию границ оценки при переходе к гибким методологиям (Agile, Lean и DevOps);
- 3) сформировать набор процессных метрик и представить их интерпретацию с точки зрения подсистемы управления, предложить систему показателей и агрегирующий индекс для оценки эффективности процессов ИТ-команды.

Материалы и методы

В классической теории управления и экономических науках эффективность трактуется как соотношение результата и затрат, либо как степень достижения целей при заданных ограничениях ресурсов. На уровне процессов подобная трактовка эффективности предполагает, что она задается комбинацией результативности (достижение цели, удовлетворенность потребителя), экономичности (минимизация затрат) и стабильности результата (сохранение способности производить результат в динамике). Для процессов, направленных на производство нематериальных продуктов (отражающих деятельность ИТ-команды), прямое измерение результата оказывается затруднительным, поскольку ценность проявляется с временными задержками, а качество часто обнаруживается путем выявления дефектов, фиксации инцидентов и закономерно связывается с повышением затрат на изменения (их устранение) [3]. Вследствие этого, оценка эффективности в ИТ должна включать в себя измерение свойств процесса, которые выступают опережающими индикаторами будущих результатов. К таким свойствам относятся скорость, предсказуемость потока, уровень потерь на ожидание и изменения, метрика устойчивости поставки, темп накопления технического долга, а также удельная стоимость производства изменений [4].

Рассматривая Agile как специфическое обстоятельство функционирования команды ИТ-разработчиков, задающее ряд руководствующих ценностей, важно подчеркнуть, что его практическая реализация осуществляется с использованием Lean и Flow – категорий бережливого производства. В рамках последнего при его применении в разработке программного обеспечения основное внимание уделяется именно анализу потока работ (Flow), с учетом его прохождения через определенные стадии. Например, известны случаи применения кумулятивных диаграмм потока и набора мер для отслеживания статуса и повышения пропускной способности с целью сокращения времени поставки [5]. Аналогично, по данным другого исследования, управление временем поставки ИТ-продукта в инкрементной и Agile-разработке зависит от того, каким образом будет распределено время по фазам разработки, а также от факторов очередности, передачи и изменений политики процесса [6].

Тем самым, важным фактором эффективности процессов в ИТ-команде становится ход проводимых работ, который зависит от принимаемых руководством решений. В рамках того же Kanban (метод управления рабочими процессами, относимый к Agile) команда обязательно отслеживает, сколько задач выполняется одновременно, старается делать задачи меньшего размера, поддерживать постоянный темп работы и обращает внимание на скопление очередности задач. Так, основные преимущества работы с Kanban сводятся к улучшению работы, к возможности точнее контролировать сроки и осознанно управлять процессами поставки продукта. Однако, с другой стороны, наиболее частые сложности связаны с тем, что командам необходимо постоянно соблюдать ограничения на количество задач в работе, учитывать зависимости между разными специалистами и договориться о единых критериях завершения задачи и ее перевода другим разработчикам. Из этого следует, что показатели потока в Канбане нужны прежде всего для управления узкими местами и ограничениями процесса, а не для формального подсчета производительности в привычном, производственном смысле.

Именно поэтому (несмотря на эффективную реализацию процессов) при оценке эффективности в Agile-среде нельзя ограничиваться скоростью процессов и выполнения задач. Важно учитывать, как быстро команда выпускает изменения и насколько стабильно работает система в целом. В практике DevOps предлагается рассматривать эффективность через четыре показателя (метрики DORA): сколько времени проходит от идеи до внедрения изменения, как часто происходят релизы, как часто изменения приводят к сбоям и сколько времени уходит на восстановление после них [7, 8]. Показатели направлены на минимизацию ситуаций, когда команда работает быстро, но при этом допускает ошибки, тратит много времени на исправления (и в итоге принцип предсказуемости нарушается), что приводит к перерасходу ресурсов.

Отдельного внимания заслуживает технический долг – это накопленные упрощения и «временные решения» в коде и архитектуре, которые часто возникают при попытках ускорить работу в краткосрочной перспективе. Сам по себе технический долг неизбежен, однако проблемы начинаются при отказе от осознанного управления им. В таком случае изменения становятся дороже, сроки менее предсказуемыми, а развитие системы замедляется [9]. Практика показывает, что инженерные меры, например, проверка кода, автоматическое тестирование и непрерывная интеграция, напрямую влияют на то, как быстро накапливается технический долг и насколько управляемыми остаются процессы [10]. Поэтому при оценке эффективности важно учитывать не только скорость, но и объем изменений, показатели технического долга, поскольку именно они зачастую становятся скрытыми ограничителями эффективности ИТ-команды.

Учитывая данные обстоятельства, настоящее исследование направлено на обобщение исследований по Agile, SPI в agile-контекстах, метрикам потока и DevOps-метрикам с последующей разработкой концептуальной модели и процедуры расчета агрегирующего индекса оценки эффективности. Модель ориентирована на данные, доступные в большинстве современных ИТ-организаций:

- трекер задач (Jira/YouTrack/Redmine) со статусами, датами переходов, типами задач;
- репозиторий (Git) с коммитами, мердж-реквестами;
- CI/CD со сборками, деплоями, результатами тестов;
- мониторинг и инциденты с событиями деградации, временем восстановления;
- инструменты качества кода (при наличии), например, code smells, debt ratio;
- учет времени и/или финансов (при наличии) со стоимостью команды, расходами на инфраструктуру.

Для построения модели оценки эффективности процессов в ИТ-команде при применении Agile предлагается руководствоваться следующим алгоритмом-последовательностью действий:

- 1) определение единицы анализа, а именно продукта или потока изменений, границ команды, типов элементов работы (feature, defect, tech debt, incident);
- 2) согласование процессных точек измерения – единые определения начала и завершения для времени поставки и времени цикла, правила фиксации «готово» и «поставлено»;
- 3) расчет базовых метрик потока и устойчивости, среди которых lead time, cycle time, throughput, WIP, PCE, CV, change failure rate, MTTR и др.;
- 4) расчет метрик потерь – доли переработок, дефектов, индексы технического долга (прокси);
- 5) нормирование показателей для сопоставимости;
- 6) агрегирование в индексы и общий показатель I_{eff} .

Так, новизна предлагаемого подхода состоит в построении агрегирующего индекса I_{eff} , который одновременно учитывает поток работ, устойчивость поставки и экономические потери, а также задает процедуру нормирования и интерпретации, привязанную к конкретным действиям команды разработчиков. В качестве проверяемых предположений выделяются: (Н1) снижение lead time достигается при одновременном контроле WIP и вариативности потока; (Н2) ускорение поставки без контроля устойчивости (change failure rate, mean time to restore) повышает долю переработок и ускоряет накопление технического долга; (Н3) интегральная оценка по трем направлениям позволяет раньше выявлять деградацию процесса по сравнению с результатными KPI и корректировать улучшения.

В рамках исследования важно указать, что применение Agile в целом видоизменяет общий подход к оценке эффективности (Таблица 1), поскольку:

- во-первых, оценка перестает быть укрупненной, т. к. вместо кварталов или года она проводится постоянно по итогам отдельных итераций и выпусков изменений;
- во-вторых, меняется сам объект анализа, поскольку рассматривается продукт и непрерывный поток изменений, которые в него вносятся (вместо проекта);
- в-третьих, данные для оценки все меньше собираются вручную в виде отчетов и все больше извлекаются из цифровых следов работы команды через системы управления задачами, репозитории кода и инструменты автоматизации.

Таблица 1 – Изменение подхода к оценке эффективности при переходе к Agile
Table 1 – Changes in the approach to efficiency evaluation when transitioning to Agile

Элемент оценивания	Традиционный подход	Agile/Lean/DevOps-подход
Объект	проект/этап	продукт и поток изменений
Целевой результат	выполнение плана	регулярная поставка ценности
Фокус метрик	«срок-бюджет-объем»	поток, качество и его устойчивость
Частота	квартал/год	спринт/неделя/непрерывно
Источник данных	отчеты	инструменты жизненного цикла
Основные риски	перерасход бюджета, просрочка	вариативность потока, инциденты, технический долг
Реакция руководства	корректировка плана	устранение ограничений потока и потерь

Опираясь на приведенные в Таблице 1 изменения, отметим, что эффективность Agile-разработки зависит от использования согласованных словарей метрик и правил

интерпретации, иначе сопоставимость и практическая применимость оценки снижаются. Следовательно, модель должна задавать, как минимум, набор базовых метрик, правила расчета, механизм агрегирования и примеры принимаемых решений.

При переходе от оценки результата к оценке процесса целевая функция процесса может быть задана как «минимизировать время поставки и вариативность при ограничении на деградацию качества и рост потерь». Соответственно, процессные метрики являются опережающими, сигнализируют о проблеме до того, как она проявится. В качестве метрик процесса и ответных действий на них (принимаемые решения) вводятся следующие обстоятельства (Таблица 2):

Таблица 2 – Метрики процесса в Agile-команде
Table 2 – Process metrics in an Agile IT team

Группа	Метрика (показатель)	Содержание, расчет	Принимаемое решение
Flow	Lead time (I_{LT})	время от запроса до поставки	устранение стадий ожидания
Flow	Cycle time (I_{CY})	время активной работы	декомпозиция задач, уменьшение очередей
Flow	Throughput (I_{TP})	завершенные элементы/период	баланс спроса и пропускной способности
Flow	WIP (I_{WIP})	незавершенная работа	WIP-лимиты; выравнивание потока
Variability	CV (throughput) (I_{CV})	$CV = \sigma/\mu$	снижение вариативности, batch size (размер пакета)
Lean	Flow efficiency (I_{PCE})	VA time/lead time	сокращение ожидания и передач
Quality	Defect escape (I_{Qdef})	дефекты после релиза	shift-left (раннее тестирование), критерии приемки
DevOps	Change failure rate (I_{CFR})	доля релизов с инцидентами	стабилизация пайплайна
DevOps	MTTR (I_{MTTR})	среднее время восстановления	наблюдаемость, runbooks
Cost efficiency	Cost per unit / effort (I_{Cunit})	затраты ресурсов на единицу поставленной ценности	оптимизация структуры затрат, ограничение «дорогого» ускорения
Loss	Rework ratio (I_{Rework})	доля исправлений/переделок	устранение причин дефектов
Debt	Debt proxy ($I_{DebtCost}$)	code smells/долг-задачи	планирование погашения

Так, метрики DORA используются как измеримые показатели того, насколько быстро и стабильно команда поставляет изменения. Основная ценность данной области заключается в том, что такие показатели можно автоматически рассчитывать на основе данных из инструментов разработки, тем самым отказаться от ручных отчетов и субъективных оценок. При их использовании важно соблюдать два принципиальных

правила. Во-первых, показатели нельзя превращать в личные KPI отдельных сотрудников, т. к. они предназначены для оценки работы процесса в целом. Во-вторых, скорость всегда должна рассматриваться вместе с устойчивостью поставки. Например, если время внесения изменений сокращается, но при этом возрастает число сбоя после релизов – значит, проблемы не исчезли, а просто «переехали» в другую зону, в инциденты, в аварии и дополнительные изменения и исправления. В итоге, как было отмечено ранее, формальная скорость возрастает, а реальная эффективность процесса снижается.

Исходя из предложенных метрик процесса, предлагается нормировать показатели в диапазон [0;1] и агрегировать их по группам – операционные, качество/устойчивость, экономические, с опорой на мультипликативную форму для отражения эффекта влияния лимитирующего фактора.

Так, частный интегральный индекс операционной эффективности рассчитывается по следующей формуле:

$$I_{oper} = (I_{LT} \cdot I_{CY} \cdot I_{TP} \cdot I_{VAR} \cdot I_{PCE})^{\frac{1}{5}}. \quad (1)$$

Показатель I_{VAR} рассчитывается по следующей формуле:

$$I_{VAR} = I_{WIP} \cdot I_{CV}. \quad (2)$$

Частный интегральный индекс качества и устойчивости процессов рассчитывается по формуле:

$$I_{qual} = (I_{Qdef} \cdot I_{CFR} \cdot I_{MTTR})^{\frac{1}{3}}. \quad (3)$$

Частный интегральный индекс экономической эффективности рассчитывается по формуле:

$$I_{econ} = (I_{Cunit} \cdot I_{Rework} \cdot I_{DebtCost})^{\frac{1}{3}}. \quad (4)$$

Интегральный индекс процессной эффективности рассчитывается по формуле:

$$I_{eff} = (I_{oper}^{\alpha} \cdot I_{qual}^{\beta} \cdot I_{econ}^{\gamma})^{\frac{1}{\alpha+\beta+\gamma}}, \quad (5)$$

где α, β, γ – весовые коэффициенты частных интегральных индексов, отражающие относительную значимость соответствующих компонентов (при отсутствии априорных оснований принимаются равными).

Определение весовых коэффициентов допустимо с использованием нескольких различных способов. В первую очередь возможно экспертное формирование весов на основе процедур субъективного распределения значимости критериев, т. к. в такого рода подходах веса выводятся из нормированных экспертных оценок относительной важности критериев. К числу наиболее применимых процедур относятся SMART и SWING, в рамках которых сначала выполняется ранжирование критериев, затем выбирается опорный критерий, после чего остальные критерии оцениваются относительно него. Данный способ представляется особенно удобным в случаях, когда веса необходимо связать со стратегическими приоритетами развития продукта, поскольку он позволяет отразить представление о том, какие свойства процесса являются более критичными в конкретном контексте развития продукта [11].

При необходимости более формализованного согласования весовых коэффициентов может использоваться процедура попарного сравнения критериев, в частности, Analytic Hierarchy Process (АНР). АНР предполагает иерархическое представление задачи, последовательное попарное сопоставление критериев по шкале

относительной важности и последующее вычисление вектора приоритетов на основе матрицы сравнений. Существенным достоинством данного подхода является наличие проверки согласованности экспертных суждений, поэтому АНР целесообразен в тех случаях, когда необходимо не только задать веса, но и обосновать их [12].

Также весовые коэффициенты могут определяться исходя из целевых обязательств по обслуживанию продукта, выраженных через SLA/OLA-приоритеты. Данный подход широко используется в работе команд, которые специализируются на разработке платформ, инфраструктуры и связанных сервисов, т. к. для них метрики устойчивости и качества объективно более значимы. Так, достижение более высоких SLA-целей связано с непропорциональным повышением операционных и технологических затрат; итоговый же уровень SLA определяется совокупностью подкомпонентов и зависимостей сервиса. Следовательно, при SLA-ориентированном формировании весовых коэффициентов допустимо повышать вклад соответствующих индексов по сравнению с операционными метриками потока, если именно соблюдение сервисных обязательств является основным ограничением процесса [13].

Нормирование в рамках предложенного индекса осуществляется двумя возможными способами: 1) по скользящему окну из 6–12 итераций, 2) по целевым порогам зрелости. Для показателей типа «меньше – лучше» расчет осуществляется по формуле (6), для показателей типа «больше – лучше» по формуле (7). Нормирование показателей осуществляется с использованием усеченной квантильной min-max трансформации (5–95 %), что позволяет снизить влияние выбросов и обеспечить сопоставимость метрик в составе интегрального индекса.

$$I_x = \begin{cases} 1 \\ \frac{q_{0,95}-x}{q_{0,95}-q_{0,05}}, x \leq q_{0,05}, q_{0,05} < x < q_{0,95}, x \geq q_{0,95}. \\ 0 \end{cases} \quad (6)$$

$$I_x = \begin{cases} 0 \\ \frac{x-q_{0,05}}{q_{0,95}-q_{0,05}}, x \leq q_{0,05}, q_{0,05} < x < q_{0,95}, x \geq q_{0,95}. \\ 1 \end{cases} \quad (7)$$

При малом числе проектов нормирование выполняется по коридорам целевых значений, задаваемым на основе SLA, внутренних нормативов или уровней процессной зрелости. В этом случае используется таргет-ориентированная кусочно-линейная нормализация, которая делает возможной интерпретируемость индекса при ограниченной выборке, что рассчитывается по формулам (8), (9) для показателей типа «меньше – лучше» и «больше – лучше» соответственно:

$$I_x = \begin{cases} 1 \\ \frac{T_{max}-x}{T_{max}-T_{min}}, x \leq T_{min}, T_{min} < x < T_{max}, x \geq T_{max}. \\ 0 \end{cases} \quad (8)$$

$$I_x = \begin{cases} 0 \\ \frac{x-T_{min}}{T_{max}-T_{min}}, x \leq T_{min}, T_{min} < x < T_{max}, x \geq T_{max}. \\ 1 \end{cases} \quad (9)$$

Так, полученные в результате нормирования и агрегирования значения частных и интегрального индексов интерпретируются в диапазоне [0;1]; значения, которые приближаются к «1», соответствуют более высокой эффективности соответствующего компонента процесса, а значения, близкие к «0», указывают на наличие выраженных ограничений и потерь. При этом принципиально важно, что интерпретация индексов

носит сравнительный и диагностический характер, а не используется для фиксации абсолютных нормативов эффективности.

Частные интегральные индексы позволяют локализовать источник деградации процессной эффективности. Так, снижение I_{oper} при стабильных значениях I_{qual} и I_{econ} свидетельствует о проблемах организации потока работ (перегрузка WIP, высокая вариативность, неэффективная декомпозиция). В свою очередь, снижение I_{qual} при сохранении высокой операционной скорости указывает на деградацию устойчивости поставки и рост потерь, связанных с инцидентами и восстановлением. Ухудшение I_{econ} отражает рост удельных затрат, переработок и накопление технического долга, которые ограничивают возможности масштабирования процессов.

При этом интегральный индекс процессной эффективности I_{eff} отражает согласованное состояние процесса по трем направлениям и позволяет отслеживать динамику эффективности во времени. Использование мультипликативной формы агрегирования принципиально исключает компенсацию низких значений одного компонента за счет высоких значений других, тем самым удается фиксировать влияние лимитирующих факторов и обеспечивать корректную интерпретацию результатов. В конечном счете реализуется цикл измерения, диагностики и принятия решения, что соответствует идеям разработки по принципам Agile.

Результаты

Апробация выполнена на выгрузке данных работы трех продуктовых команд (Далее – Проект А, Проект В и Проект С), которые применяли Agile-подходы и делали регулярные релизы. Выбранный интервал наблюдения составил 8 недель (4 итерации по 2 недели). Источником данных послужил трекер задач (даты переходов по статусам, WIP, типы работ), CI/CD (устанавливался факт деплоев, сбой релизов), мониторинг и инциденты (время восстановления), учет затрат (стоимость команды как OpEx, приведенная к показателю на единицу поставки).

В соответствии с представленной ранее моделью эффективность оценивается как агрегирование трех индексов; отдельно также стоит указать, что ритмичность операционализируется через вариативность throughput (коэффициент вариации CV), поскольку чем ниже CV, тем выше ритмичность (в индексе отражено компонентом I_{VAR}).

Так, исходные агрегированные показатели по проектам (ИТ-командам) представлены в Таблице 3. Отметим, что в рамках апробации показатель технического долга использовался как прокси-индикатор интенсивности накопленных инженерных обязательств в наблюдаемом периоде. Расчет Debt proxy выполнялся как средневзвешенная величина двух компонент. Первая – debt ratio по данным инструмента статического анализа кода, определялась как отношение оценочного объема работ по устранению code smells, поддерживаемых нарушений и иных дефектов к оценочному объему разработки анализируемой базы кода; вторая – доли задач типа tech debt/refactoring/architecture improvement в общем потоке задач за рассматриваемый интервал. В формализованном виде показатель Debt proxy рассчитывался по формуле:

$$Debtproxy = 0,5 \cdot TDcode + 0,5 \cdot TDtask, \quad (10)$$

где TDcode – debt ratio, %, TDtask – доля debt-задач в общем потоке, %.

Равные веса компонент выбраны по причине ограниченного интервала наблюдения и необходимости одновременно учитывать как структурные характеристики кодовой базы, так и фиксируемую потребность в погашении долга через backlog. Соответственно, значения Debt proxy, приведенные далее в Таблице 3, отражают относительную интенсивность его накопления и проявления в исследуемом периоде.

Таблица 3 – Агрегированные метрики эффективности, собранные за 8 недель
Table 3 – Aggregated efficiency metrics collected over an 8-week period

Показатель	Проект А	Проект В	Проект С
Lead time, дней (медиана)	7,5	14,0	9,5
Cycle time, дней (медиана)	4,0	6,5	4,8
Throughput, элементов/нед	28	22	25
WIP, среднее	12	26	18
CV(Throughput)	0,22	0,40	0,30
PCE (Process Cycle Efficiency)	0,35	0,18	0,28
Defect escape, дефектов на 100 элементов	2,0	3,5	6,0
Change failure rate (CFR), %	8	12	22
MTTR, часов	2,5	4,0	6,5
Стоимость единицы поставки (relative)	1,00	1,15	1,10
Rework, % от потока	10	18	25
Debt proху, % (доля/индекс техдолга)	12	20	28

Как можно заметить, содержательно формируются различные профили эффективности команд; по Проекту А фиксируются: короткий lead time, низкая вариативность, высокий PCE, низкие дефекты и быстрый recovery; по проекту В большой WIP, высокая вариативность, низкий PCE, т. е. ожидания и очереди влияют на поток; по проекту С операционный поток находится на приемлемом уровне, однако устойчивость и качество ниже (высокие CFR, дефекты и rework), что транслируется в экономические потери.

Чтобы избежать эффекта «краевых значений» при малом числе проектов, нормирование выполнено не по min-тах трех наблюдений, а по коридорам целевых значений. Каждая метрика переводится в индекс $[0;1]$, где 1 – «лучшее» значение (например, минимальный lead time, минимальный CFR, максимальный PCE). Далее рассчитываются индексы как геометрическое среднее компонент (по заданным ранее формулам). Результаты расчета индексов и сопоставление проектов сгруппированы в Таблице 4:

Таблица 4 – Результаты расчета индексов
Table 4 – Results of index calculations

Индекс	Проект А	Проект В	Проект С
I_{oper}	0,819	0,393	0,651
I_{qual}	0,885	0,694	0,287
I_{econ}	0,856	0,497	0,317
I_{eff}	0,853	0,514	0,390

Итак, относительно низкая эффективность выявлена у ИТ-команд проектов В и С; расчет индексов позволяет отметить основные причины низкой эффективности. В случае Проекта В проблемы выявлены в операционных вопросах деятельности; в случае Проекта С – в качестве и устойчивости, а также экономических аспектах разработки.

В случае Проекта А установлена более высокая эффективность по причине сбалансированной эффективности.

Кроме того, в результатах апробации находит свое подтверждение гипотеза H1 о том, что снижение времени поставки достигается при одновременном контроле WIP и вариативности потока. В частности, по Проекту А выявлен наименьший lead time при одновременном низком среднем уровне WIP и минимальном значении коэффициента вариации throughput, что является следствием ритмичности и управляемости потока

работ. В Проекте В, напротив, высокий WIP и повышенная вариативность сопровождаются существенным повышением lead time и снижением операционного индекса I_{oper} . Тем самым подтверждается, что само по себе увеличение пропускной способности без ограничения незавершенной работы и контроля вариативности не приводит к ускорению поставки, а напротив, усиливает эффекты ожидания и очередей.

Результаты также подтверждают гипотезу H2, согласно которой ускорение поставки без контроля устойчивости приводит к увеличению переработок и накоплению технического долга. В Проекте С операционные показатели находятся на приемлемом уровне, однако высокие значения change failure rate и MTTR сопровождаются увеличением дефектов после релизов, доли rework и индекса технического долга, что отражается в резком снижении индексов I_{qual} и I_{econ} , несмотря на относительно сбалансированный поток. Таким образом, эмпирически фиксируется перенос потерь из поставки в инциденты, с исправлениями и экономическими издержками, что подтверждает теоретическое положение о неэффективности ускорения за счет качества.

Гипотеза H3 о преимуществе интегральной оценки по трем направлениям по сравнению с результатными KPI также находит подтверждение, т. к. анализ отдельных метрик (например, throughput или lead time) не позволяет однозначно интерпретировать состояние процессов в Проектах В и С, поскольку отдельные показатели могут находиться на приемлемом уровне. В то же время интегральный индекс I_{eff} позволяет установить снижение эффективности за счет лимитирующих факторов и потенциально делает возможной локализацию проблемной области еще до того, как негативные эффекты проявятся в сроках, бюджете или удовлетворенности заказчика. Тем самым подтверждается, что процессно-ориентированная интегральная оценка обладает более высоким диагностическим и прогностическим потенциалом по сравнению с традиционными KPI, ориентированными на результат.

Обсуждение

Предложенная модель оценки эффективности процессов ИТ-команды при применении гибких методологий управления обосновывает переход от результатных KPI к процессной системе измерений, которая будет отражать поток работ, устойчивость поставки и экономические последствия. Исследования по измерению потока и lead time подтверждают, что сокращение времени поставки и повышение прозрачности достигаются через управление WIP, throughput и вариативностью [5, 6], а при использовании Kanban основным является связывание метрик с действиями руководителей по устранению ограничений потока [4]. Метрики DevOps (DORA) позволяют сохранять баланс между скоростью и устойчивостью, и допускают автоматизацию на данных инструментов разработки и CI/CD [7, 8]. Компонента технического долга и переработок встраивается в качестве ограничителя масштабирования и долгосрочной эффективности, что подтверждается практико-ориентированными исследованиями [9, 10].

Кроме того, предлагаемый подход соотносится с рядом известных систем измерения. Например, близкой выступает упомянутая ранее DORA-перспектива, которая ориентирована прежде всего на оценку скорости и устойчивости поставки и действительно пригодна для автоматизированного мониторинга software delivery performance. Вместе с тем, метрики стабильности определены менее однозначно, чем скоростные, а также не охватывают все значимые аспекты качества; кроме того, эффекты технического долга могут проявляться лишь в средне- и долгосрочном горизонте. В этом смысле DORA-подход удобен для оценки delivery performance, но в базовой конфигурации не выделяет технический долг как отдельную компоненту анализа [7].

Интересным также представляется SPACE-framework; по данным научной литературы, продуктивность разработки не может быть сведена к одному показателю или одному измерению, т. к. должна, напротив, рассматриваться как многомерная. В оценку по SPACE-framework целесообразно включение: satisfaction and well-being, performance, activity, communication and collaboration, efficiency and flow. Тем самым SPACE-framework шире оценки собственно процессной эффективности команды разработки, т. к. охватывает не только характеристики потока и результата, но и состояние участников, коммуникационные аспекты и организационный контекст. Его ценность особенно высока при диагностике developer productivity, однако для мониторинга на уровне одной продуктовой команды (с короткими интервалами и автоматизацией) он излишне трудоемкий, поскольку предполагает использование смешанных данных (опросные и организационно-поведенческие показатели) [14].

Помимо этого, схожими являются flow- и value-stream-подходы, широко используемые в software engineering. Они ориентированы прежде всего на визуализацию и анализ сквозного движения работы через процесс разработки. По данным исследований, value stream mapping и FLOW позволяет выявлять и систематизировать проблемы информационного и процессного потока в крупномасштабной разработке, что позволяет характеризовать процесс и выделять его узкие места. Следовательно, подобные подходы перспективно применять для end-to-end-анализа потока, выявления задержек, передач и потерь в системе разработки в целом и т. п. Вместе с тем их основная сила состоит именно в картировании и улучшении value stream [15]; предложенный же интегральный индекс позволяет одновременно учитывать поток, устойчивость поставки, переработки, экономические потери и прокси-показатель технического долга. В этом смысле предложенная в статье модель занимает промежуточное положение, поскольку в целом сохраняет чувствительность flow-подходов к состоянию процессов и измеримость DevOps-метрик, и в то же время дополняет их экономическим и ценными с позиции управления показателями, что делает возможным сопоставление команд одного класса и мониторинг изменений на коротких интервалах.

Практическая ценность модели заключается в том, что она, во-первых, задает минимальный набор метрик, который будет применим в большинстве ИТ-команд; во-вторых, предлагает удобный для применения нормируемый индекс, отражающий три стороны эффективности; в-третьих, обеспечивает интерпретацию показателей через типовые действия руководства и команды (устранение узких мест потока, стабилизация поставки, планирование погашения долга).

Тем самым модель становится инструментом операционного управления (управление потоком и качеством) и тактического управления (приоритизация улучшений и перераспределение ресурсов).

С практической точки зрения важно и то, что модель допускает ее внедрение в формате регламента. В минимальной конфигурации целесообразно закрепить:

- 1) владельца метрик (руководитель разработки / engineering manager);
- 2) ответственного за сбор и валидацию данных (аналитик; DevOps; РМО, что зависит от организационной структуры);
- 3) площадку интерпретации и принятия решений.

Артефактами модели выступают дашборды базовых метрик и индексов, карточки причин отклонений (в виде гипотезы, подтверждающих ее показателей и выявляемых категорий риска), перечень улучшений с указанием ожидаемого механизма влияния на индексы (I_{oper} , I_{qual} и I_{econ}) и срока проверки результата. Формализация модели позволит повышать управляемость процесса, снижать зависимость от субъективной экспертной оценки и обеспечивать сопоставимость состояний команды по итерациям.

Отдельная возможность внедрения модели состоит и в экономической интерпретации результатов, поскольку компоненты, связанные с переделками, дефектами, инцидентами и техническим долгом, могут быть приведены к стоимостной оценке через показатели трудоемкости и стоимости часа работы команды разработчиков, а затем использованы для обоснования приоритетов улучшений и оценки результатов управления.

В результате модель выполняет функцию связующего звена между инженерными метриками и показателями эффективности, востребованными на уровне управления, что позволяет обосновывать внутренние изменения через измеримый результат.

Итак, практическая ценность модели заключается в том, что она, во-первых, задает минимальный набор метрик, который будет применим в большинстве ИТ-команд; во-вторых, предлагает удобный для применения нормируемый индекс, отражающий три стороны эффективности; в-третьих, обеспечивает интерпретацию показателей через типовые действия руководства и команды (устранение узких мест потока, стабилизация поставки, планирование погашения долга).

Основным ограничением исследования является его концептуальный характер в части предложенной модели; имеет место необходимость эмпирической валидации с проверкой чувствительности к различным типам работ и масштабам команд, что найдет отражение в будущих исследованиях автора.

Заключение

В статье предложена процессно-ориентированная модель оценки эффективности деятельности ИТ-команд в условиях применения гибких методологий управления, основанная на агрегировании метрик потока работ, устойчивости поставки и экономических потерь. В рамках исследования показано, что традиционные результатные KPI не позволяют адекватно отражать специфику Agile-разработки и своевременно выявлять деградацию процессов, в то время как использование процессных метрик, опирающихся на цифровые следы жизненного цикла ИТ-продукта, обеспечивает более корректную и интерпретируемую оценку эффективности.

Результаты экспериментальной апробации модели на данных ИТ-команд продемонстрировали ее практическую применимость и диагностический потенциал. Было установлено, что снижение времени поставки достигается при одновременном контроле незавершенной работы и вариативности потока, а попытки ускорения без обеспечения устойчивости поставки приводят к увеличению числа дефектов, переработок и накоплению технического долга. Показано, что команды с сопоставимыми операционными характеристиками могут существенно различаться по уровню устойчивости и экономической эффективности, что не выявляется при использовании отдельных метрик скорости или пропускной способности.

Кроме того, полученные результаты подтвердили, что интегральный индекс процессной эффективности, сформированный по трем направлениям, позволяет раньше выявлять проблемные зоны по сравнению с результативными показателями. Использование мультипликативной формы агрегирования исключает компенсацию низкого качества или высокой аварийности за счет повышения скорости и тем самым отражает влияние лимитирующих факторов процесса, что, в свою очередь, делает возможной локализацию источников неэффективности и формирование адресных мероприятий, направленных на выравнивание потока, стабилизацию поставки и снижение экономических потерь.

Практическая значимость полученных результатов заключается в возможности применения предложенной модели в управлении Agile-командами для мониторинга

состояния процессов и обоснования мероприятий по их улучшению. В перспективе дальнейшие исследования могут быть направлены на расширение эмпирической базы, уточнение целевых диапазонов оцениваемых метрик для команд различной зрелости, а также на апробацию модели в условиях масштабных портфелей ИТ-проектов и распределенных команд.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Dybå T., Dingsøy T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*. 2008;50(9–10):833–859. <https://doi.org/10.1016/j.infsof.2008.01.006>
2. Cornide-Reyes H., Madrigal G., Muñoz G., et al. Analysis of the use of software process improvement models in agile development. *Ingeniare. Revista chilena de ingeniería*. 2024;32. <https://doi.org/10.4067/S0718-33052024000100213>
3. Menezes R., Marinho M., Sampaio S. Metrics for Large-Scale Agile Development: A Survey of the Brazilian Software Industry. In: *Proceedings of the XXIII Brazilian Symposium on Software Quality (SBQS 2024), 05–08 November 2024, Salvador, Bahia, Brazil*. New York: ACM; 2024. P. 210–219. <https://doi.org/10.1145/3701625.3701660>
4. Dos Santos P.S.M., Beltrão A.C., de Souza P.D., Travassos G.H. On the benefits and challenges of using kanban in software engineering: A structured synthesis study. *Journal of Software Engineering Research and Development*. 2018;6(1). <https://doi.org/10.1186/s40411-018-0057-1>
5. Petersen K., Wohlin C. Measuring the flow in Lean software development. *Software: Practice and Experience*. 2011;41(9):975–996. <https://doi.org/10.1002/spe.975>
6. Petersen K. An empirical study of lead-times in incremental and agile software development. In: *New Modeling Concepts for Today's Software Processes: International Conference on Software Process (ICSP 2010), 08–09 July 2010, Paderborn, Germany*. Berlin, Heidelberg: Springer; 2010. P. 345–356. https://doi.org/10.1007/978-3-642-14347-2_30
7. Sallin M., Kropp M., Anslow C., Quilty J.W., Meier A. Measuring software delivery performance using the four key metrics of DevOps. In: *Agile Processes in Software Engineering and Extreme Programming: 22nd International Conference on Agile Software Development (XP 2021), 14–18 June 2021, Virtual Event*. Cham: Springer; 2021. P. 103–119. https://doi.org/10.1007/978-3-030-78098-2_7
8. Rügger J., Kropp M., Graf S., Anslow C. Fully Automated DORA Metrics Measurement for Continuous Improvement. In: *Proceedings of the 2024 International Conference on Software and Systems Processes (ICSSP 2024), 04–06 September 2024, Munich, Germany*. New York: ACM; 2024. <https://doi.org/10.1145/3666015.3666020>
9. Lim E., Taksande N., Seaman C. A balancing act: what software practitioners have to say about technical debt. *IEEE Software*. 2012;29(6):22–27. <https://doi.org/10.1109/MS.2012.130>
10. Holvitie J., Licorish Sh.A., Spínola R.O., et al. Technical debt and agile software development practices and processes: An industry practitioner survey. *Information and Software Technology*. 2018;96:141–160. <https://doi.org/10.1016/j.infsof.2017.11.015>
11. Ezell B., Lynch Ch.J., Hester P.T. Methods for weighting decisions to assist modelers and decision analysts: A review of ratio assignment and approximate techniques. *Applied Sciences*. 2021;11(21). <https://doi.org/10.3390/app112110397>
12. Gompf K., Traverso M., Hetterich J. Using analytical hierarchy process (AHP) to introduce weights to social life cycle assessment of mobility services. *Sustainability*. 2021;13(3). <https://doi.org/10.3390/su13031258>

13. Lucz G., Forstner B. Optimizing service level agreement tier selection in online services through legacy lifecycle profile and support analysis: A quantitative approach. *Mathematics*. 2025;13(11). <https://doi.org/10.3390/math13111743>
14. Forsgren N., Storey M.-A.D., Maddila Ch.Sh., et al. The SPACE of developer productivity: There's more to it than you think. *ACM Queue*. 2021;19(1):20–48.
15. Ali N.B., Petersen K., Schneider K. FLOW-assisted value stream mapping in the early phases of large-scale software development. *Journal of Systems and Software*. 2016;111:213–227. <https://doi.org/10.1016/j.jss.2015.10.013>

ИНФОРМАЦИЯ ОБ АВТОРЕ / INFORMATION ABOUT THE AUTHOR

Котова Милана Римовна, заместитель вице-президента, старший ИТ-бизнес-аналитик, Инвестиционный банк Барклайс, Прага, Чешская Республика. **Milana R. Kotova**, Assistant Vice President, Senior IT Business Analyst, Barclays Investment Bank, Prague, Czech Republic.
e-mail: milanark2015@gmail.com
ORCID: [0009-0005-9135-4916](https://orcid.org/0009-0005-9135-4916)

Статья поступила в редакцию 11.02.2026; одобрена после рецензирования 19.03.2026; принята к публикации 11.05.2026.

The article was submitted 11.02.2026; approved after reviewing 19.03.2026; accepted for publication 11.05.2026.