

Эмпирическая валидация многоуровневой когнитивной архитектуры командной разработки программного обеспечения

Н. Ф. Тулкунова*, Д. Суччи

АНО ВО «Университет Иннополис», г. Иннополис, Российская Федерация

Адрес: 420500, Российская Федерация, г. Иннополис, ул. Университетская, д. 1

* n.tulkunova@innopolis.ru

Аннотация

Современная разработка программного обеспечения (ПО) представляет собой форму коллективной интеллектуальной деятельности, в которой значительная часть проектных решений формируется в процессе взаимодействия участников команды. В данной работе представлен эмпирический анализ трансформации представлений участников команд разработки, возникающих в ходе командного обсуждения требований, архитектурных решений и реализации системы. В качестве концептуальной основы исследования используется многоуровневая когнитивная архитектура командной разработки ПО, связывающая когнитивные процессы участников, используемые проектные артефакты и фазы жизненного цикла разработки (SDLC). Эмпирическое исследование основано на наблюдении пяти команд разработки ПО и анализе 20 эпизодов коллективного взаимодействия. В ходе анализа было выявлено 42 когнитивных перехода, то есть случаев изменения представлений участников команды о проектных решениях, зафиксированных в коммуникации и проектных артефактах. Для систематизации выявленных переходов предложена их типология, включающая согласование, уточнение, реконструкцию и усиление представлений. Показано, что распределение типов переходов зависит от фаз жизненного цикла разработки и выполняемых участниками задач: на фазах инженерии требований и проектирования доминируют переходы согласования и уточнения, тогда как на фазах реализации и тестирования возрастает роль реконструкции и усиления. Полученные результаты уточняют динамический аспект предложенной когнитивной архитектуры и служат эмпирической валидацией её ключевых предположений о характере когнитивных изменений в командах разработки ПО. Результаты позволяют рассматривать процесс разработки ПО как последовательность когнитивных переходов, возникающих в ходе коллективного взаимодействия, и создают основу для дальнейших исследований и инструментальной поддержки командной разработки.

Ключевые слова: когнитивная архитектура, командная разработка программного обеспечения, распределённая когнитивная, когнитивные переходы, проектные артефакты, жизненный цикл разработки программного обеспечения

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Тулкунова Н. Ф., Суччи Д. Эмпирическая валидация многоуровневой когнитивной архитектуры командной разработки программного обеспечения // Современные информационные технологии и ИТ-образование. 2026. Т. 22, № 1. С. 106-116. <https://doi.org/10.25559/SITITO.022.202601.106-116>

© Тулкунова Н. Ф., Суччи Д., 2026



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Empirical Validation of a Multilevel Cognitive Architecture for Collaborative Software Development

N. F. Tulkunova*, G. Succi

Innopolis University, Innopolis, Russian Federation

Address: 1 Universitetskaya St., Innopolis 420500, Russian Federation

* n.tulkunova@innopolis.ru

Abstract

Modern software development represents a form of collective intellectual activity in which many project decisions emerge through interactions among team members. This study presents an empirical analysis of changes in developers' representations that occur during collaborative discussions of requirements, architectural decisions, and implementation strategies. The study is grounded in a multilevel cognitive architecture of collaborative software development that links developers' cognitive processes, project artifacts, and the phases of the Software Development Life Cycle (SDLC). The empirical investigation is based on observations of five software development teams and the analysis of 20 episodes of collaborative interaction. The analysis identified 42 cognitive transitions, defined as changes in team members' understanding of design and implementation decisions, traced through team communication and project artifacts. To systematize the observed changes, a typology of cognitive transitions is proposed, including alignment, refinement, reconstruction, and reinforcement of representations. The results show that the distribution of transition types varies across different phases of the development process and depends on the tasks and roles of team members: alignment and refinement dominate during requirements engineering and design, while reconstruction and reinforcement become more prominent during implementation and testing. These findings refine the dynamic aspect of the proposed cognitive architecture and provide empirical validation of its core assumptions about cognitive change in software development teams. They suggest that the software development process can be understood as a sequence of cognitive transitions emerging through collaborative interaction and lay the groundwork for future research and tool support for collaborative software engineering.

Keywords: cognitive architecture, shared understanding, distributed cognition, macrocognition, collaborative software engineering, Software Development Life Cycle, cognitive artifacts

Conflict of interests: The authors declare no conflict of interest.

For citation: Tulkunova N.F., Succi G. Empirical Validation of a Multilevel Cognitive Architecture for Collaborative Software Development. *Modern Information Technologies and IT-Education*. 2026;22(1):106-116. <https://doi.org/10.25559/SITITO.022.202601.106-116>



Введение

Современная разработка ПО является не только инженерным процессом создания программных систем, но и формой коллективной интеллектуальной деятельности [12-14]. Значительная часть проектных решений формируется в ходе взаимодействия участников команды [3-5]. В условиях гибких методологий разработки и распределённых команд разработчики постоянно вовлечены в процессы анализа требований, обсуждения архитектурных решений, согласования способов реализации и анализа возникающих ошибок [7], [11]. Эти процессы требуют не только технической компетентности участников, но и формирования общего понимания проектных решений, обеспечивающего согласованность действий команды. Исследования в области программной инженерии всё чаще обращаются к когнитивным аспектам командной разработки, рассматривая деятельность разработчиков как совокупность взаимосвязанных когнитивных процессов, включающих интерпретацию требований, формирование архитектурных решений, планирование реализации и мониторинг результатов разработки [3, 4]. В ряде работ показано, что значительная часть сложностей разработки ПО связана не столько с техническими аспектами реализации, сколько с различиями в понимании требований, архитектурных решений и целей проекта между участниками команды [17-20].

Одним из теоретических направлений, позволяющих анализировать такие процессы, является подход распределённого познания (*distributed cognition*), в рамках которого когнитивная деятельность рассматривается как распределённая система, включающая не только участников команды, но и используемые ими инструменты, документы и другие внешние представления знаний [6], [8-10]. В контексте разработки ПО это означает, что понимание требований, архитектурных решений и планов реализации формируется и поддерживается не только в индивидуальном сознании разработчиков, но и в проектных артефактах – пользовательских историях, архитектурных диаграммах, коде, документации и системах управления задачами [15, 16].

В последние годы в исследованиях программной инженерии активно изучаются процессы формирования общего понимания требований (*shared understanding of requirements*) и согласования решений в командах разработки [1], [19], [21]. В частности, Бьюкен [1] предложил модель развития общего понимания требований, в которой коллективное обсуждение рассматривается как последовательность переходов между когнитивными состояниями участников команды. Эти переходы возникают в результате обсуждений, уточнений и устранения различий в интерпретации требований.

Несмотря на развитие подобных подходов, большинство существующих исследований сосредоточено на отдельных аспектах разработки ПО – например, на процессе формирования требований или на когнитивной нагрузке разработчиков [4, 5]. При этом значительно реже анализируется динамика когнитивных изменений участников команды на протяжении различных фаз SDLC. Например, недостаточно изучено, каким образом коллективное взаимодействие разработчиков приводит к изменениям их представлений о требованиях, архитектуре системы или способах реализации задач на различных этапах разработки.

Для анализа подобных процессов ранее авторами была предложена многоуровневая когнитивная архитектура командной разработки ПО, описывающая взаимодействие когнитивных процессов разработчиков, используемых проектных артефактов и фаз SDLC [2]. В данной архитектуре выделяются три взаимосвязанных уровня: уровень когнитивной активности участников разработки, уровень внешних представлений и проектных артефактов, а также уровень временной структуры фаз разработки. Такое представление позволяет рассматривать командную разработку ПО как систему, в которой когнитивные процессы участников команды поддерживаются и фиксируются в артефактах разработки и разворачиваются во времени по мере продвижения проекта. Однако предложенная архитектура носит преимущественно концептуальный характер и требует эмпирической проверки применимости для анализа реальных процессов командной разработки. В частности, остаётся открытым вопрос о том, как именно изменяются представления участников команды о проектных решениях в ходе коллективного взаимодействия и какие типы когнитивных изменений возникают на различных фазах SDLC.

В настоящей работе предлагается рассматривать такие изменения как когнитивные переходы – изменения представлений участников команды о требованиях, архитектурных решениях или способах реализации задач, возникающие в результате коллективного взаимодействия и фиксируемые в проектных артефактах и коммуникации участников. Анализ таких переходов позволяет описывать динамику формирования коллективного понимания в командах разработки и выявлять типичные формы когнитивной трансформации, возникающие в ходе разработки ПО.

Цель настоящего исследования состоит в эмпирической валидации динамического компонента многоуровневой когнитивной архитектуры командной разработки программного обеспечения. Под динамическим компонентом понимается описываемая архитектурой структура когнитивных переходов между исходными и итоговыми представлениями участников команды, связанных с ключевыми когнитивными функциями на различных фазах SDLC.

В контексте данной работы эмпирическая валидация означает сопоставление предсказаний архитектуры о типах когнитивных переходов и их связи с фазами SDLC и ролями участников с результатами анализа наблюдений за реальными командами. Модель считается частично или полностью валидированной, если наблюдаемые типы переходов и их распределение по фазам разработки и ролям согласуются с ожидаемыми паттернами.

Для достижения поставленной цели решаются следующие задачи:

1. Операционализировать ключевые элементы многоуровневой когнитивной архитектуры для эмпирического анализа, задав наблюдаемые показатели когнитивных состояний и когнитивных переходов в контексте командной разработки ПО;
2. Собрать и проанализировать данные о работе пяти проектных команд, зафиксировав эпизоды коллективного взаимодействия и связанные с ними изменения представлений участников о требованиях, архитектуре, планах реализации и поведении системы;
3. Выделить и классифицировать когнитивные переходы, наблюдаемые в ходе этих эпизодов, по типам (согласование, уточнение, реконструкция, усиление) и фазам SDLC;



4. Сопоставить наблюдаемое распределение типов когнитивных переходов по фазам разработки и ролям участников с ожиданиями, заложенными в многоуровневую когнитивную архитектуру, и обсудить степень и характер этого соответствия;

5. Обсудить ограничения проведённой эмпирической валидации и наметить направления дальнейших исследований, направленных на уточнение и расширение архитектуры.

Эмпирическая часть исследования основана на наблюдении пяти команд разработки и включает анализ коллективных обсуждений, интервью с участниками и проектных артефактов. Это позволяет эмпирически проверить применимость предложенной архитектуры для анализа командной разработки.

В дальнейшем под когнитивным состоянием участника или команды понимается совокупность их актуальных представлений о требованиях, архитектуре, планах реализации и поведении системы, реконструируемых на основе высказываний, совместных обсуждений и артефактов разработки.

Под когнитивным переходом понимается наблюдаемое изменение когнитивного состояния, зафиксированное через трансформацию формулировок требований или архитектурных решений, изменение структуры и содержания проектных артефактов (задачи, диаграммы, код, отчёты о дефектах), а также через явное пересмотренное объяснение поведения системы в процессе коллективного обсуждения.

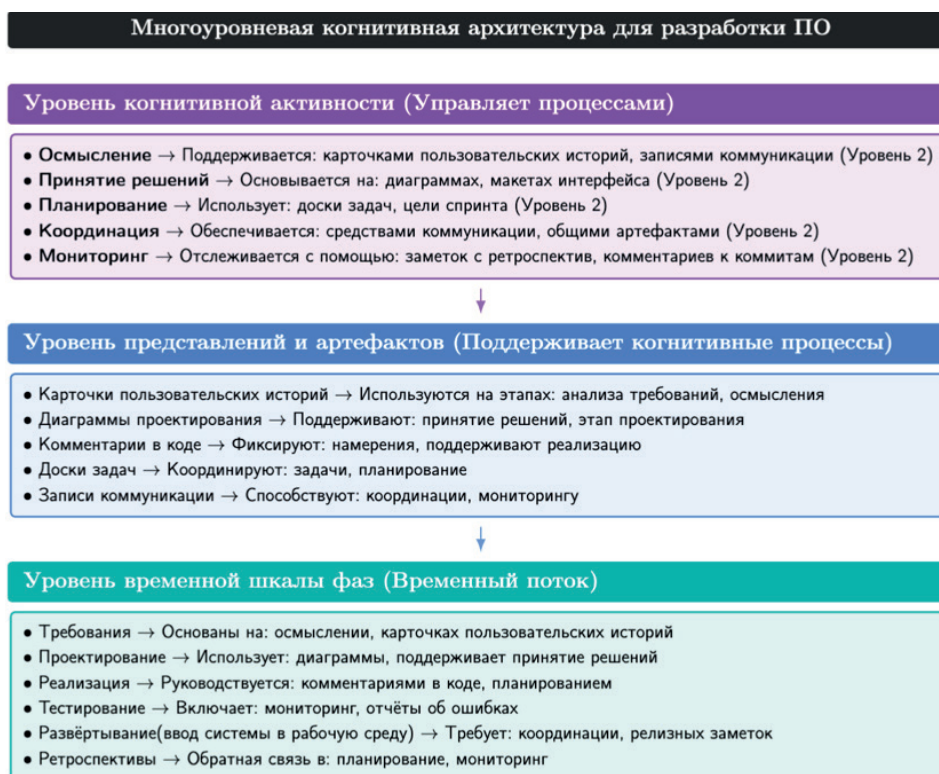
Концептуальная модель исследования

Для анализа трансформации представлений участников команд разработки используется многоуровневая когнитивная архитектура командной разработки ПО, предложенная в предыдущей работе авторов [2]. Данная архитектура рассматривает разработку ПО как распределённую когнитивную систему, в которой когнитивные процессы участников команды поддерживаются внешними представлениями и развиваются во времени по мере продвижения проекта.

В архитектуре выделяются три взаимосвязанных уровня (рис. 1). Первый уровень – когнитивная активность – включает основные когнитивные процессы участников разработки, такие как осмысление задач, принятие решений, планирование, координация и мониторинг. Эти процессы отражают коллективную когнитивную деятельность команды при решении задач разработки.

Второй уровень – представления и артефакты – включает внешние средства поддержки когнитивных процессов, такие как пользовательские истории, архитектурные диаграммы, доски задач, комментарии в коде и записи коммуникации. Эти артефакты выступают в роли когнитивных опор, позволяя участникам команды фиксировать и передавать знания, а также координировать командную деятельность.

Третий уровень – временная структура разработки – отражает развитие когнитивных процессов во времени и связывает их с фазами SDLC, включая анализ требований, проектирование, реализацию, тестирование и ретроспективу.



Р и с. 1. Многоуровневая когнитивная архитектура командной разработки ПО

F i g. 1. A multilevel cognitive architecture for collaborative software development

Источник: здесь и далее в статье все таблицы и рисунки составлены авторами.

Source: Hereinafter in this article all tables and figures were made by the authors.



В рамках настоящего исследования данная архитектура используется как концептуальная модель для анализа коллективных эпизодов разработки. Предполагается, что изменения представлений участников команды проявляются в виде когнитивных переходов, возникающих в процессе коллективного взаимодействия и фиксируемых в проектных артефактах. Таким образом, данная архитектура задаёт аналитическую рамку исследования, в которой коллективные эпизоды разработки рассматриваются как ситуации возникновения и фиксации когнитивных переходов.

Материалы и методы

В качестве методологической основы исследования использовался качественный эмпирический подход, позволяющий анализировать процессы выработки решений в реальном контексте командной разработки ПО. Такой подход широко применяется в исследованиях когнитивных процессов разработки ПО и позволяет изучать взаимодействие разработчиков, коммуникацию участников и эволюцию проектных артефактов [3], [5]. В рамках исследования когнитивные процессы разработки рассматриваются как распределённые между участниками команды и используемыми ими внешними представлениями – проектными артефактами и инструментами разработки. Такое понимание согласуется с подходом распределённого познания, согласно которому когнитивная деятельность формируется во взаимодействии между людьми, инструментами и внешними представлениями информации [6], [8]. Методология исследования ориентирована на анализ коллективных эпизодов разработки, в которых формируются и уточняются проектные решения. Каждый такой эпизод рассматривался как ситуация коллективного взаимодействия, включающая исходные представления участников, процесс обсуждения и итоговые договорённости, зафиксированные в проектных артефактах.

Участники исследования и контекст разработки

В исследовании приняли участие пять команд разработки программного обеспечения, работающих по гибким методологиям (Scrum или Kanban) в коммерческих компаниях различного профиля. Каждая команда включала 5–8 участников (разработчики, аналитики, UX-специалисты, тестировщики, технические лидеры) и использовала сходный набор инструментов совместной разработки (системы управления задачами, средства совместного документирования и прототипирования, системы контроля версий). Такой контекст позволил исследовать когнитивные процессы в условиях реальной командной разработки, сохраняя при этом сопоставимость ролей участников и типов используемых артефактов.

В процессе работы команды использовали различные проектные артефакты и средства коммуникации, включая:

- системы управления задачами (Jira, Trello);
- системы командной документации и визуализации (Confluence, Miro, Figma);
- системы управления версиями и кодовой базой (GitHub, GitLab).

Эти инструменты использовались для фиксации требований, обсуждения архитектурных решений, координации задач и анализа проблем, возникающих в ходе разработки.

Сбор эмпирических данных

Сбор данных осуществлялся в форме полевых наблюдений за процессом работы команд разработки и анализа их проектных артефактов. В исследовании использовалось сочетание нескольких источников данных:

1. Полуструктурированные наблюдения. Фиксировались эпизоды коллективного взаимодействия, относящиеся к ключевым фазам SDLC: обсуждение требований, архитектурные совещания, совместная разработка и обсуждение дефектов. Наблюдения проводились в естественных условиях (рабочие встречи, созвоны, стендапы), с последующим восстановлением хода обсуждения по заметкам и цифровым следам (протоколы встреч, комментарии в системах управления задачами);
2. Контекстные интервью. До и после выбранных эпизодов проводились короткие интервью с участниками (35–45 минут), направленные на реконструкцию их исходных и итоговых представлений о требованиях, архитектурных решениях и поведении системы. Интервью позволяли зафиксировать то, что не всегда явно проговаривается на встречах, но влияет на интерпретацию обсуждаемых решений;
3. Анализ проектных артефактов. Для каждого эпизода анализировались связанные артефакты: пользовательские истории и задачи в системах управления задачами, архитектурные диаграммы, фрагменты кода и комментарии в системах контроля версий, отчёты о дефектах и тестовые сценарии. Изменения этих артефактов рассматривались как материальные следы когнитивных переходов (уточнение требований, пересмотр архитектурных решений, изменение планов реализации).

В качестве проектных артефактов анализировались:

- пользовательские истории и задачи в системах управления задачами;
- архитектурные схемы и диаграммы;
- комментарии и сообщения в системах управления версиями;
- отчёты о дефектах и тестовые сценарии;
- записи коммуникации участников команды.

Использование нескольких источников данных позволило сопоставлять изменения представлений участников команды с изменениями проектных артефактов и обсуждениями, происходящими в процессе разработки.

Наблюдения проводились в течение нескольких итераций разработки для каждой команды. Общая продолжительность наблюдений составила около 50 часов. Дополнительно было проведено 14 полуструктурированных интервью с участниками команд. Средняя продолжительность одного интервью составляла 35–45 минут.

Единицы анализа

В качестве единицы анализа рассматривались эпизоды коллективного взаимодействия участников разработки, в ходе которых обсуждались и уточнялись проектные решения. Например, в качестве таких эпизодов рассматривались:

- обсуждение требований и пользовательских историй;
- архитектурные обсуждения;
- совместную разработку кода;
- анализ дефектов и тестовых сценариев.



Всего в ходе исследования было выделено 20 эпизодов коллективного взаимодействия. Каждый эпизод анализировался как последовательность трёх этапов:

1. **исходное состояние**, отражающее первоначальное понимание задачи участниками команды;
2. **коллективное взаимодействие**, включающее обсуждение и согласование решений;
3. **итоговое состояние**, зафиксированное в проектных артефактах и высказываниях участников.

При анализе эпизодов учитывался контекст соответствующей фазы разработки и связанных с ней когнитивных процессов.

Предполагалось, что на различных фазах разработки доминируют различные когнитивные процессы участников команды, поддерживаемые определёнными проектными артефактами. Для систематизации наблюдений была сформирована схема соответствия между фазами разработки, когнитивными процессами и используемыми проектными артефактами. Такая схема представлена в Таблице 1. Она отражает операционализацию уровней предложенной многоуровневой когнитивной архитектуры, связывая когнитивные процессы участников разработки, используемые проектные артефакты и фазы разработки.

Таблица 1. Когнитивные процессы и артефакты на различных фазах разработки
Table 1. Cognitive processes and artifacts at different phases of SDLC

Фаза разработки	Основной когнитивный процесс	Примеры коллективных эпизодов	Наблюдаемые артефакты
Инженерия требований	Осмысление	обсуждение пользовательских историй	пользовательские истории
Проектирование	Планирование и принятие решений	архитектурные обсуждения	архитектурные диаграммы
Реализация	Координация действий	совместная разработка кода	код и комментарии
Тестирование	Мониторинг	анализ дефектов	отчёты о дефектах
Ретроспектива	Рефлексия	обсуждение результатов спринта	записи ретроспектив

Представленная схема позволила определить наблюдаемые индикаторы когнитивных процессов на различных фазах разработки и использовать проектные артефакты как источник данных для анализа изменений представлений участников команды.

Процедура анализа данных

Анализ собранных данных проводился поэтапно. Для систематизации процесса анализа была сформирована процедура сопоставления исходного и итогового понимания проектных решений участниками команды. Основные этапы анализа и используемые источники информации представлены в Таблице 2.

Таблица 2. Процедура анализа данных
Table 2. Data analysis procedure

Этап анализа	Источник данных	Цель анализа
Фиксация исходного состояния	интервью, существующие артефакты	определить первоначальное понимание задачи
Анализ коллективного взаимодействия	наблюдение обсуждений	выявить аргументы и процессы согласования
Фиксация итогового состояния	обновлённые артефакты	определить итоговое проектное решение
Сопоставление состояний	сравнение исходных и итоговых данных	выявить когнитивные переходы

На первом этапе фиксировалось исходное понимание задачи на основе интервью и существующих артефактов. На втором анализировались коллективные обсуждения и аргументация решений. На третьем этапе определялось итоговое состояние решения, зафиксированное в обновлённых артефактах.

Классификация когнитивных переходов

Выявленные случаи трансформации понимания проектных решений рассматривались как когнитивные переходы – изменения представлений участников команды о требованиях, архитектурных решениях или способах реализации задач, возникающие в результате коллективного взаимодействия.

Для систематизации наблюдаемых изменений использовалась типология когнитивных переходов, представленная в Таблице 3. Выделенные типы переходов отражают различные формы трансформации представлений участников команды при коллективной работе.

Таблица 3. Типы когнитивных переходов
Table 3. Types of cognitive transitions

Тип перехода	Описание
Согласование	устранение расхождений между представлениями участников
Уточнение	детализация существующих представлений
Реконструкция	пересмотр исходного понимания задачи
Усиление	подтверждение существующего понимания



Следует отметить, что в анализе данные пяти команд рассматривались в совокупности. Такой подход оправдан тем, что все команды работали в сходном контексте гибкой разработки, использовали аналогичные роли и инструменты и выполняли сопоставимые по структуре задачи. В то же время результаты описывают общие паттерны когнитивных переходов и не претендуют на выявление различий между отдельными командами или организационными контекстами.

Результаты

В соответствии с описанной процедурой анализа для каждого эпизода коллективного взаимодействия сопоставлялись исходное и итоговое понимание проектных решений участниками команды.

Эмпирический анализ был основан на изучении 20 эпизодов коллективного взаимодействия, зафиксированных в пяти командах разработки ПО. Каждый эпизод рассматривался как последовательность трёх этапов: исходного понимания задачи, коллективного обсуждения и итогового состояния, зафиксированного в проектных артефактах.

Анализ эпизодов позволил выявить изменения представлений участников команды о требованиях, архитектурных решениях и способах реализации задач. Эти изменения рассматривались как когнитивные переходы, возникающие в результате коллективного взаимодействия участников разработки.

В ходе анализа было выявлено 42 когнитивных перехода, возникающих на различных фазах SDLC.

Когнитивные переходы на фазе инженерии требований

Эпизоды, относящиеся к фазе инженерии требований, включали обсуждения пользовательских историй, уточнение требований и согласование функциональности системы.

До начала обсуждений участники команды часто имели различное понимание требований. Эти различия проявлялись в интерпретации пользовательских сценариев, терминологии и приоритетов функциональности. Например, в одном из эпизодов обсуждения требований разработчики по-разному интерпретировали поведение системы при обработке пользовательского запроса. В ходе обсуждения было выявлено расхождение в понимании требований, после чего была сформулирована новая пользовательская история, уточняющая ожидаемую функциональность системы. Этот случай был отнесён к переходу типа согласования.

В процессе коллективного обсуждения происходило уточнение формулировок требований и согласование их интерпретации между участниками команды. Итоговые решения фиксировались в пользовательских историях и системах управления задачами.

Для фазы инженерии требований наиболее характерными оказались переходы согласования и уточнения. Участники постепенно формировали общее понимание требований, уточняя их содержание и устраняя различия в интерпретации.

Когнитивные переходы на фазе проектирования

На фазе проектирования анализировались эпизоды обсуждения архитектурных решений и выбора способов реализации функциональности системы.

Предварительные интервью показали, что участники команды часто формировали первоначальные представления об архитектуре системы на основе частичного контекста задачи. Эти представления могли различаться по уровню детализации и используемым архитектурным решениям.

Коллективные архитектурные обсуждения приводили к формированию согласованной архитектурной схемы системы. В ходе обсуждений участники команды обменивались аргументами, анализировали альтернативные решения и корректировали свои представления о структуре системы.

Для фазы проектирования преобладали переходы согласования и реконструкции, когда первоначальные представления участников пересматривались или объединялись в новое архитектурное решение.

Итоговые решения фиксировались в архитектурных диаграммах, схемах и документации проекта.

Когнитивные переходы на фазе реализации

Эпизоды, относящиеся к фазе реализации, включали наблюдения за совместной разработкой кода и обсуждением способов реализации функциональности.

На этой фазе основное внимание участников команды было направлено на координацию действий и согласование технических деталей реализации. В ходе командной работы разработчики обсуждали структуру кода, способы реализации функций и возможные альтернативные решения.

В большинстве наблюдаемых эпизодов когнитивные изменения носили характер уточнения. Разработчики постепенно уточняли детали реализации и согласовывали способы решения задач.

В некоторых случаях наблюдались переходы типа усиления, когда первоначальное понимание задачи подтверждалось в процессе реализации.

Итоговые решения фиксировались непосредственно в коде и комментариях в системах управления версиями.

Когнитивные переходы на фазе тестирования

На фазе тестирования анализировались эпизоды обсуждения обнаруженных дефектов и анализа поведения системы. Обнаружение ошибок или отклонений от ожидаемого поведения системы часто приводило к пересмотру первоначальных представлений участников команды о работе системы. В этих ситуациях происходили когнитивные переходы типа реконструкции, когда участники команды пересматривали свои представления о причинах возникновения дефектов или о структуре системы.

В других случаях наблюдались переходы типа усиления, когда тестирование подтверждало корректность ранее принятых решений.

Результаты анализа фиксировались в отчётах о дефектах и тестовых сценариях.

Распределение когнитивных переходов

В ходе анализа было зафиксировано 42 когнитивных перехода. Распределение переходов по типам представлено в Таблице 4 и отражает суммарные результаты анализа 20 эпизодов коллективного взаимодействия.



Таблица 4. Распределение когнитивных переходов
Table 4. Distribution of cognitive transitions

Тип перехода	Количество	Доля
Согласование	16	38 %
Уточнение	12	29 %
Реконструкция	8	19 %
Усиление	6	14 %

Как и предполагает многоуровневая когнитивная архитектура, наибольшую долю составили переходы согласования и уточнения (в сумме 67%), то есть случаи устранения расхождений и детализации существующих представлений о проектных решениях. Это подтверждает гипотезу о том, что ключевым механизмом формирования коллективного понимания требований и решений в командах разработки являются именно процессы согласования и поэтапного уточнения представлений.

Распределение переходов по фазам разработки

Анализ показал, что различные фазы разработки характеризуются различными типами когнитивных переходов, что соответствует предположениям многоуровневой когнитивной архитектуры о связи когнитивных функций с фазами SDLC. Распределение переходов по фазам представлено в Таблице 5.

Таблица 5. Когнитивные переходы на различных фазах разработки
Table 5. Cognitive transitions at different phases of SDLC

Фаза разработки	Наиболее характерные переходы
Инженерия требований	согласование, уточнение
Проектирование	согласование, реконструкция
Реализация	уточнение, усиление
Тестирование	реконструкция, усиление

На фазе инженерии требований доминируют переходы согласования и уточнения, что отражает переход от разрозненных индивидуальных интерпретаций к разделяемому пониманию пользовательских историй и критериев приёмки. На фазе проектирования более заметна роль реконструкции, когда первоначальные архитектурные представления пересматриваются в ходе обсуждений. В фазе реализации преобладают переходы уточнения и усиления, связанные с детализацией решений в коде и подтверждением ранее принятых архитектурных выборов. На стадии тестирования реконструкция и усиление отражают либо пересмотр представлений о поведении системы при обнаружении дефектов, либо подтверждение корректности принятых решений. Таким образом, характер когнитивных изменений систематически зависит от фазы разработки и выполняемых участниками задач, что эмпирически подтверждает динамический компонент архитектуры. Анализ эпизодов коллективного взаимодействия также показал различия в типах когнитивных переходов, инициируемых участниками команды различных ролей. Разработчики чаще

инициировали переходы типа уточнения, связанные с согласованием и детализацией технических решений на уровне реализации. Аналитики и UX-специалисты чаще инициировали реконструкцию представлений на этапе обсуждения требований, когда переосмысливались сценарии использования системы. Технические лидеры демонстрировали преимущественно переходы типа усиления, что согласуется с их ролью в формировании и поддержании целостного архитектурного видения. Тестировщики играли ключевую роль в инициировании реконструкции на фазе тестирования, поскольку обнаружение дефектов требовало пересмотра представлений о работе системы. Эти наблюдения конкретизируют, как различные роли в команде реализуют предусмотренные архитектурой когнитивные функции на разных фазах SDLC.

В целом результаты показывают, что коллективное взаимодействие систематически сопровождается изменениями в понимании проектных решений, проявляющимися в виде когнитивных переходов разных типов. Наиболее распространёнными оказываются переходы согласования и уточнения, обеспечивающие формирование общего понимания требований и способов реализации задач, тогда как реконструкция и усиление связаны с ситуациями повышенной неопределённости или проверкой ранее принятых решений.

Обсуждение

Полученные результаты позволяют уточнить динамический компонент многоуровневой когнитивной архитектуры командной разработки ПО. Эмпирический анализ 20 эпизодов коллективного взаимодействия показал, что изменения представлений участников о проектных решениях действительно проявляются в виде когнитивных переходов, распределённых по типам и фазам SDLC так, как предполагает архитектура. Это позволяет рассматривать командную разработку ПО как процесс когнитивной координации, в ходе которого представления участников систематически трансформируются в результате коллективного взаимодействия.

Выявленные типы когнитивных переходов – согласование, уточнение, реконструкция и усиление – отражают различные формы изменения представлений участников разработки и конкретизируют, как коллективное взаимодействие влияет на формирование общего понимания требований и решений. Данный результат согласуется с результатами исследований, посвящённых формированию общего понимания требований (*shared understanding of requirements*) в командах разработки ПО [1], [23].

Полученные результаты также подтверждают положения теории распределённого познания, согласно которой когнитивные процессы в социотехнических системах распределены между участниками деятельности и используемыми ими внешними представлениями [8], [10]. В контексте разработки ПО такими внешними представлениями выступают проектные артефакты – пользовательские истории, архитектурные диаграммы, код и системы управления задачами. Анализ эпизодов коллективного взаимодействия показал, что именно через такие артефакты происходит фиксация и стабилизация коллективного понимания проектных решений.

Результаты исследования также демонстрируют фазо-специфический характер когнитивных переходов. На фазе инжене-



рии требований преобладают переходы согласования и уточнения, связанные с устранением различий в интерпретации требований. На фазе проектирования чаще наблюдаются переходы реконструкции, возникающие при пересмотре первоначальных архитектурных решений. На фазе реализации основную роль играют переходы уточнения, связанные с детализацией технических решений, тогда как на фазе тестирования чаще происходят переходы реконструкции и усиления, возникающие при анализе поведения системы и обнаружении дефектов.

Такая фазовая специфика когнитивных переходов указывает на то, что различные этапы разработки активируют различные когнитивные процессы участников команды. Это наблюдение согласуется с результатами исследований когнитивных аспектов разработки ПО, показывающих, что задачи анализа требований, проектирования и реализации требуют различных типов когнитивной деятельности разработчиков [3, 4].

Полученные результаты позволяют также уточнить предложенную ранее многоуровневую когнитивную архитектуру командной разработки ПО [2]. В рамках этой архитектуры когнитивные процессы участников разработки рассматриваются в тесной связи с используемыми проектными артефактами и фазами разработки. Эмпирический анализ показал, что изменения представлений участников команды действительно проявляются на пересечении этих уровней: когнитивные процессы участников разработки реализуются через коллективное взаимодействие и фиксируются в проектных артефактах, которые затем используются на последующих этапах разработки.

Таким образом, когнитивные переходы можно рассматривать как механизм, связывающий когнитивную активность участников разработки с эволюцией проектных артефактов и продвижением разработки по фазам жизненного цикла. Такое представление позволяет рассматривать процесс разработки ПО как последовательность когнитивных преобразований коллективного понимания системы.

Практическая значимость полученных результатов связана с возможностью использования концепции когнитивных переходов для анализа процессов коллективного взаимодействия в командах разработки. Выявление ситуаций, в которых происходят различные типы когнитивных переходов, может использоваться для анализа эффективности командных обсуждений, выявления проблем в интерпретации требований и оценки процессов согласования архитектурных решений.

Кроме того, предложенный подход может быть использован при разработке инструментов поддержки командной разработки ПО. Например, анализ изменений проектных артефактов и коммуникации участников команды может позволить выявлять ситуации, требующие дополнительного обсуждения или уточнения требований.

При интерпретации полученных результатов следует учитывать ограничения проведённого исследования. Во-первых, исследование основано на наблюдении ограниченного числа команд разработки ПО, что не позволяет делать статистически обобщённые выводы о распространённости выявленных типов когнитивных переходов. Во-вторых, анализ носит качественный характер и основан на интерпретации наблюдаемых эпизодов коллективного взаимодействия. Тем не менее наблюдение нескольких независимых команд разработки позволило выявить повторяющиеся паттерны когнитивных из-

менений, возникающих в процессе командной разработки ПО. Тем самым предложенный подход создаёт основу для операционализации анализа коллективного познания в командах разработки через наблюдаемые когнитивные переходы. Дальнейшие исследования могут быть направлены на расширение эмпирической базы анализа, включая наблюдение большего числа команд разработки и использование количественных методов анализа данных. Перспективным направлением также является разработка методов автоматизированного анализа проектных артефактов и коммуникации участников команды для выявления когнитивных переходов в процессе разработки ПО.

Заключение

В работе выполнена эмпирическая проверка применимости многоуровневой когнитивной архитектуры для анализа командной разработки ПО. Цель исследования заключалась в выявлении и классификации когнитивных переходов – изменений представлений участников команды о требованиях, архитектурных решениях и способах реализации задач, возникающих на различных фазах SDLC.

Исследование опиралось на наблюдение пяти команд и анализ 20 эпизодов коллективного взаимодействия, в рамках которых было выявлено 42 когнитивных перехода.

В ходе исследования была предложена и применена типология когнитивных переходов, включающая четыре основных типа: согласование, уточнение, реконструкция и усиление. Полученные результаты показали, что наиболее распространёнными являются переходы согласования и уточнения, обеспечивающие формирование общего понимания требований и способов реализации задач в командах разработки.

Анализ распределения когнитивных переходов по фазам разработки показал, что различные этапы разработки характеризуются различными типами когнитивных изменений. На фазе инженерии требований преобладают переходы согласования и уточнения, связанные с устранением различий в интерпретации требований. На фазе проектирования чаще наблюдаются переходы реконструкции, возникающие при пересмотре архитектурных решений. Фаза реализации характеризуется переходами уточнения, связанными с детализацией технических решений, тогда как на фазе тестирования чаще происходят переходы реконструкции и усиления, возникающие при анализе поведения системы.

Полученные результаты позволяют рассматривать процесс разработки ПО как последовательность когнитивных переходов, возникающих в ходе коллективного взаимодействия участников команды. Такой подход позволяет уточнить динамический аспект предложенной ранее многоуровневой когнитивной архитектуры командной разработки ПО, связывая когнитивные процессы участников разработки, проектные артефакты и фазы жизненного цикла разработки.

Практическая значимость исследования заключается в возможности использования концепции когнитивных переходов для анализа процессов коллективного взаимодействия в командах разработки ПО. Выявление ситуаций, в которых происходят различные типы когнитивных переходов, может способствовать более глубокому пониманию процессов формирования общего понимания требований и согласования архитектурных решений в командах разработки.



Следует отметить, что проведённое исследование имеет ряд ограничений. Во-первых, анализ основан на наблюдении ограниченного числа команд разработки ПО. Во-вторых, исследование носит качественный характер и основано на интерпретации наблюдаемых эпизодов коллективного взаимодействия.

В дальнейшем планируется расширение эмпирической базы

исследования за счёт анализа большего числа команд разработки и различных типов проектов. Перспективным направлением дальнейших исследований является разработка методов автоматизированного анализа проектных артефактов и коммуникации участников команды, позволяющих выявлять когнитивные переходы и поддерживать когнитивную согласованность в процессе разработки.

References

- [1] Buchan J. An empirical cognitive model of the development of shared understanding of requirements. In: Requirements Engineering. In: D. Zowghi, Z. Jin (eds.). Berlin; Heidelberg: Springer; 2014. p. 165-179. https://doi.org/10.1007/978-3-662-43610-3_13
- [2] Tulkunova N.F., Succi G. Multilevel cognitive architecture for collaborative software development across all stages of the software development lifecycle (SDLC). *Information Science and Control Systems*. 2026;(1):86-97. (In Russ., abstract in Eng.) https://doi.org/10.22250/18142400_2026_87_1_86
- [3] Peitek N., Siegmund J., Apel S., Kästner C., Parnin C., Bethmann A., Leich T., Saake G., Brechmann A. A look into programmers' heads. *IEEE Transactions on Software Engineering*. 2018;46(4):442-462. <https://doi.org/10.1109/TSE.2018.2863303>
- [4] Gonçalves L.J., Farias K., da Silva B.C. Measuring the cognitive load of software developers: An extended systematic mapping study. *Information and Software Technology*. 2021;136:106563. <https://doi.org/10.1016/j.infsof.2021.106563>
- [5] Xu S., Rajlich V. Dialog-based protocol: an empirical research method for cognitive activities in software engineering. In: Proceedings of the International Symposium on Empirical Software Engineering. 2005. p. 10. <https://doi.org/10.1109/ISESE.2005.1541848>
- [6] Hollan J., Hutchins E., Kirsh D. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction*. 2000;7(2):174-196. <https://doi.org/10.1145/353485.353487>
- [7] Ciancarini P., Farina M., Masyagin S., Succi G., Yermolaieva S., Zagvozkina N. Non-verbal communication in software engineering: an empirical study. *IEEE Access*. 2021;9:71942-71953. <https://doi.org/10.1109/ACCESS.2021.3075983>
- [8] Lanubile F., Ebert C., Prikladnicki R., Vizcaino A. Collaboration tools for global software engineering. *IEEE Software*. 2010;27(2):52-55. <https://doi.org/10.1109/MS.2010.39>
- [9] Sharp H., Giuffrida R., Melnik G. Information flow within a dispersed agile team: A distributed cognition perspective. In: Agile Processes in Software Engineering and Extreme Programming. Berlin, Heidelberg: Springer; 2012. p. 62-76. https://doi.org/10.1007/978-3-642-30350-0_5
- [10] Sharp H., Robinson H., Segal J., Furniss D. The role of story cards and the wall in XP teams: a distributed cognition perspective. *AGILE 2006*. IEEE Press; 2006. p. 75-84. <https://doi.org/10.1109/AGILE.2006.56>
- [11] Jones P.H., Chisalita C. Cognition and collaboration: analyzing distributed community practices for design. In: CHI EA '05. New York: ACM; 2005. p. 2120. <https://doi.org/10.1145/1056808.1057116>
- [12] Meyer A.N., Barr E.T., Bird C., Zimmermann T. Today was a good day: The daily life of software developers. *IEEE Transactions on Software Engineering*. 2019;47(5):863-880. <https://doi.org/10.1109/TSE.2019.2904957>
- [13] Fernandes R.G., da Silva L.F., Vils L. Distributed team cognition and collaborative problem-solving in project management. *International Journal of Managing Projects in Business*. 2023;16(6-7):713-742. <https://doi.org/10.1108/IJMPB-05-2023-0100>
- [14] Hyysalo J., Lehto J., Aaramaa S., Kelanti M. Supporting cognitive work in software development workflows. In: Product-Focused Software Process Improvement. Berlin, Heidelberg: Springer; 2013. p. 20-34. https://doi.org/10.1007/978-3-642-39259-7_5
- [15] Murtazina M.Sh., Avdeenko T.V. The ontology-driven approach to support the requirements engineering process in scrum framework. In: Proceedings of IV International Conference on Information Technology and Nanotechnology (ITNT-2018). Samara: Novaya tekhnika; 2018. p. 2610-2620. (In Russ., abstract in Eng.) EDN: UONRCV
- [16] Batovrin V.K. Requirements engineering process in the system lifecycle. In: Technologies for the Development of Information Systems TRIS-2016: Conference Proceedings. Vol. 1. Gelendzhik: Southern Federal University; 2016. p. 26-43. (In Russ., abstract in Eng.) EDN: WKYPSP
- [17] Fernandes R.G., Vils L., da Silva L.F. Distributed cognition model for project management. *Revista Gestão & Tecnologia*. 2023;23(3):85-127. <https://doi.org/10.20397/2177-6652/2023.v23i3.2646>
- [18] Zaina L.A.M., Sharp H., Barroca L. UX information in the daily work of an agile team: A distributed cognition analysis. *International Journal of Human-Computer Studies*. 2021;147:102574. <https://doi.org/10.1016/j.ijhcs.2020.102574>
- [19] Mangalaraj G., Nerur S., Mahapatra R.K., Price K.H. Distributed cognition in software design: An experimental investigation of the role of design patterns and collaboration. *MIS Quarterly*. 2014;38(1):249-274. <https://doi.org/10.25300/MISQ/2014/38.1.12>
- [20] Xiao X., Lindberg A., Hansen S., Lyytinen K. Computing requirements for open source software: A distributed cognitive approach. *Journal of the Association for Information Systems*. 2018;19(12):2-29. <https://doi.org/10.17705/1jais.00525>
- [21] Drury-Grogan M.L. The changes in team cognition and cognitive artifact use during agile software development project management. *Project Management Journal*. 2021;52(2):127-145. <https://doi.org/10.1177/8756972820960301>

Поступила 19.02.2026; одобрена после рецензирования 27.03.2026; принята к публикации 10.04.2026.

Submitted 19.02.2026; approved after reviewing 27.03.2026; accepted for publication 10.04.2026.



Об авторах:

Тулкунова Навруза Фархадовна, аспирант факультета компьютерных и инженерных наук, АНО ВО «Университет Иннополис» (420500, Российская Федерация, г. Иннополис, ул. Университетская, д. 1), **ORCID:** <https://orcid.org/0009-0004-7436-3397>, n.tulkunova@innopolis.ru

Суччи Джанкарло, профессор факультета компьютерных и инженерных наук, АНО ВО «Университет Иннополис» (420500, Российская Федерация, г. Иннополис, ул. Университетская, д. 1), кандидат наук в сфере вычислительной техники и электротехники, **ORCID:** <https://orcid.org/0000-0001-8847-0186>, g.succi@innopolis.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Navruza F. Tulkunova, Postgraduate Student of the Faculty of Computer Science and Engineering, Innopolis University (1 Universitetskaya St., Innopolis 420500, Russian Federation), **ORCID:** <https://orcid.org/0009-0004-7436-3397>, n.tulkunova@innopolis.ru

Giancarlo Succi, Professor of the Faculty of Computer Science and Engineering, Innopolis University (1 Universitetskaya St., Innopolis 420500, Russian Federation), Candidate of Science in Computer and Electrical Engineering, **ORCID:** <https://orcid.org/0000-0001-8847-0186>, g.succi@innopolis.ru

All authors have read and approved the final manuscript.

