

# ИНТЕГРАЦИЯ СИСТЕМ УПРАВЛЕНИЯ ПРОЕКТАМИ И РАЗРАБОТКОЙ ПО

## INTEGRATION OF PROJECT MANAGEMENT AND SOFTWARE DEVELOPMENT SYSTEMS

**R. Ilyin  
S. Nikishov**

*Summary.* The paper examines approaches and solutions for integrating project management systems with software development tools. The increasingly important role of integrated platforms (Application Lifecycle Management, DevOps) for unifying planning and development processes is noted. Examples of modern tools and methods for synchronizing data between project management and development environments are provided. The advantages of such integration — improved collaboration, increased process visibility, and automated task tracking — are analyzed, as well as the challenges associated with implementing integrated solutions. It is concluded that integrating management and development systems can significantly increase the efficiency of IT project execution, while simultaneously requiring consideration of organizational changes.

*Keywords:* project management, software development, integration, Application Lifecycle Management, DevOps, collaboration, task tracking, information systems.

**Ильин Руслан Дмитриевич**

Аспирант, Российская академия народного хозяйства  
и государственной службы  
при Президенте Российской Федерации  
mildgravity0@yandex.ru

**Никишов Сергей Иванович**

д.э.н., доцент, Российская академия  
народного хозяйства и государственной службы  
при Президенте Российской Федерации  
nikishov-si@ranepa.ru

*Аннотация.* Рассматриваются подходы и решения для интеграции систем управления проектами с инструментами разработки программного обеспечения. Отмечается возросшая роль интегрированных платформ (Application Lifecycle Management, DevOps) для объединения процессов планирования и разработки. Приведены примеры современных инструментов и способов синхронизации данных между средствами управления проектами и средствами разработки. Проанализированы преимущества такой интеграции — улучшение сотрудничества, повышение прозрачности хода работ, автоматизация отслеживания задач — а также сложности, связанные с внедрением интегрированных решений. Сделаны выводы о том, что интеграция управленческих и систем управления разработкой позволяет существенно повысить эффективность реализации ИТ-проектов, одновременно требуя учета организационных изменений.

*Ключевые слова:* управление проектами, разработка программного обеспечения, интеграция, Application Lifecycle Management, DevOps, сотрудничество, отслеживание задач, информационные системы.

## Введение

В современных организациях часто используется множество разрозненных инструментов: одни для управления проектами, другие — для непосредственной разработки программного обеспечения. Традиционно процессы менеджмента проектов и процессы разработки нередко выполнялись обособленно, что приводило к недостаткам в согласованности и эффективности [1][2]. Отсутствие единого информационного пространства может вызывать дублирование ввода данных, ошибки и задержки передачи информации между менеджерами и разработчиками [2]. Согласно опросам, коммуникационные разрывы являются одним из главных препятствий успешной реализации проектов — 59 % специалистов отмечают проблемы во внутреннем взаимодействии как серьезный барьер [20]. Эти факты подчёркивают необходимость более тесной интеграции между системами управления проектами и средствами разработки ПО.

Понятие интеграции в управлении проектами отражено и в стандартах. Например, PMI определяет управление интеграцией проекта как координацию всех элементов проекта в единое целое [11]. В области разработки ПО концепция сквозного управления жизненным циклом приложений (Application Lifecycle Management, ALM) прямо направлена на объединение процессов управления, разработки и эксплуатации. Инструменты ALM обеспечивают единую стандартизированную систему для взаимодействия команд разработки ПО и связанных отделов, автоматизируя процессы сборки и доставки и повышая эффективность за счёт согласованности действий [4][5]. Таким образом, интеграция методов и систем управления проектами с инструментами разработки становится важнейшим условием успешного выполнения сложных проектов. Исследования показывают, что разрозненное выполнение управленческих и инженерных задач приводит к неэффективности и задержкам, тогда как интегрированный подход решает эти проблемы, объединяя команды, методологии и инстру-

менты в единое целое [7][8]. Интеграция усилий менеджеров проектов и разработчиков способствует сотрудничеству и снижает фрагментированность процессов [8].

Активное развитие гибких методологий и DevOps-культуры также стимулирует сближение управленческих и технических практик. Так, еще Бозм и Тёрнер отмечали необходимость баланса между строгим управлением и гибкой разработкой, предлагая комбинировать дисциплину планирования с адаптивностью agile-подходов [19][16]. В последние годы на рынке появились комплексные решения, позволяющие «под одной крышей» вести и планирование проекта, и разработку кода, и тестирование. Многие компании переходят от разрозненных инструментов (электронные таблицы, отдельные таск-трекеры) к интегрированным системам управления проектами — по мере цифровизации такие системы становятся фундаментом управления компанией [15][12]. Ниже рассмотрены существующие подходы к интеграции систем управления проектами и разработки ПО, их преимущества и проблемы внедрения.

### Инструменты и подходы к интеграции

Интегрированные платформы (ALM-системы). Один из подходов — использовать единое программное решение, охватывающее весь цикл от планирования проекта до релиза продукта. Примерами являются корпоративные платформы управления жизненным циклом приложений. Например, Microsoft Azure DevOps (ранее Team Foundation Server) объединяет средства планирования проектов, систему контроля версий, инструмент отслеживания задач и багов, а также конвейер CI/CD. Такая платформа архивирует все артефакты разработки в центральном репозитории и обеспечивает совместную работу всех внутренних и внешних участников проекта [9][14]. Благодаря этому в Azure DevOps проектные менеджеры и разработчики работают в единой среде: ведутся единые бэклоги задач, связываются изменения кода с задачами, автоматически отслеживаются сборки и развертывания. Интеграция управления проектом и разработки в TFS/Azure DevOps дает клиентам и стейкхолдерам лучшую прозрачность хода разработки, поощряя непрерывное взаимодействие заказчиков с командой разработчиков [14].

Другой пример — платформа Atlassian. Этот вендор предлагает связанный набор инструментов: Jira для управления задачами и дефектами, Confluence для документирования требований и знаний, Bitbucket (или теперь облачный Github интеграции) для управления репозиториями кода, Bamboo для сборки, а также коммуникационный модуль (ранее HipChat) [13]. В совокупности эти инструменты поддерживают как Agile-подход (спринты, бэклоги), так и классическое управление, позволяя командам вести весь цикл разработки. Jira связы-

вает изменения кода с задачами, предоставляя менеджерам проекта возможности мониторинга разработки прямо в задачах. Такой комплекс хорошо подходит для организаций, практикующих непрерывную интеграцию и DevOps; у Atlassian реализована глубокая интеграция с Git-репозиториями [13]. Как отмечено в обзоре, решения Atlassian покрывают основные потребности управления и разработки в стартапах и средних компаниях, позволяя вести требования, код и коммуникации на единой платформе [13].

Схожей интегрированной средой является GitLab, которая изначально создавалась как платформа «все в одном» для DevOps. GitLab сочетает систему управления исходным кодом (Git-репозиторий), трекер задач и багов, встроенную систему код-ревью, CI/CD конвейеры и даже средства мониторинга deployments. Благодаря этому многие команды используют GitLab не только как инструмент разработчиков, но и для управления проектными задачами. Интеграция задач и кода в GitLab позволяет менеджерам проектов отслеживать выполнение работ в реальном времени: например, видеть статус задач, прогресс пайплайнов сборки и тестирования, историю слияния веток и т. д. Инструменты вроде GitLab фактически стирают границу между «управлять проектом» и «разрабатывать проект» — все участники работают в одном пространстве. Отдельные исследования отмечают интерес использовать среду GitLab именно в задачах управления проектами, чтобы решить практические проблемы классического PM — улучшить контроль сроков, качество и коммуникацию [1][18]. Интегрируя управление проектом внутрь платформы разработки, такие системы позволяют руководителям проектов оперативно получать техническую информацию (статусы сборок, результаты проверок кода и пр.), а разработчикам — лучше понимать управленческий контекст проекта.

Интеграция посредством API и плагинов. Другой подход — связать между собой уже используемые отдельные системы управления проектами и разработки при помощи интеграционных модулей. Большинство современных корпоративных инструментов имеют открытые API или готовые коннекторы. Например, связка Jira и систем контроля версий: Jira Software содержит панель разработки (Development Panel), которая при интеграции с репозиториями отображает связанные с задачей ветки, коммиты, pull request'ы, сборки и развертывания [13]. Достаточно указать ключ задачи в сообщении коммита — и Jira автоматически привяжет этот коммит к задаче, отобразит его в интерфейсе задачи, а workflow-триггеры могут даже перевести задачу в статус «In Progress» при первом коммите [13]. Таким образом, подключив Jira к системам разработки (Bitbucket, GitHub, Bamboo и др.), команда получает сквозную прослеживаемость: менеджеры видят технический про-

гресс по каждой задаче (на каком этапе код, прошел ли сборку, деплой и т. д.), а разработчики экономят время на двойном вводе информации [16].

Аналогично, интеграция Asana с GitLab позволяет синхронизировать задачи и инциденты между этими системами. Asana выступает как удобный инструмент планирования и отслеживания задач, а GitLab — как среда разработки. При интеграции статусы задач из цикла разработки (issues, merge-requests в GitLab) автоматически отражаются в Asana, обеспечивая всем участникам единое видение прогресса [2]. Менеджеры могут в Asana в реальном времени видеть, какой код написан и когда, какие проблемы закрыты, а разработчики — получать задачи из Asana в своей привычной среде разработки [2]. По сути, достигается двухсторонняя связь отделов: через интеграцию Asana + GitLab команда налаживает оперативное общение между “бизнесом” и “разработкой”, исключая задержки из-за информационных разрывов [2]. Как отмечает DevOps-экспертиза, такая синхронизация уменьшает коммуникационные разрывы и повышает эффективность проектов [2].

Таблица 1 ниже обобщает некоторые распространенные варианты интегрированных решений и связей инструментов.

Таблица 1.

Распространенные варианты интегрированных решений и связей инструментов

Подход / Система	Характеристика интеграции
Azure DevOps (TFS)	Единая платформа ALM: задачи, репозиторий кода, сборки и деплой в одном месте [14]. Повышает прозрачность для всех участников проекта.
Atlassian (Jira+Confluence+Bitbucket)	Связанный набор инструментов: задачи связаны с требованиями и кодом, есть интеграция с Git, поддержка Agile и DevOps-практик [13].
GitLab	Полностью интегрированная DevOps-платформа: от планирования задач до CI/CD. Менеджмент проекта встроен в среду разработки (сквозное отслеживание pipeline).
Asana + GitLab	Интеграция через API: задачи Asana синхронизируются со статусами задач GitLab [2], обеспечивая единое информационное пространство.

Преимущества интеграции систем управления проектами и разработки

Интеграция инструментов управления проектами со средствами разработки дает ощутимые выигрыши, от-

мечаемые как практиками, так и исследователями. Ниже перечислены ключевые преимущества такого подхода.

- Когда управленческие и инженерные данные объединены, все участники имеют единое представление о ходе проекта [12]. Менеджер проекта в любой момент видит, какой прогресс достигнут разработчиками по каждой задаче, вплоть до конкретных коммитов и сборок [13][13]. Разработчики же понимают приоритеты и статус проекта в целом, а не только свою узкую задачу. Улучшенная видимость действий команды — один из важных эффектов интеграции Jira и GitLab, отмечаемых практиками [1].
- Интегрированная среда выступает общим коммуникационным пространством для менеджеров и разработчиков. Пропадают барьеры между «управленцами» и «технарями»: они работают с одной системой и видят обновления друг друга практически мгновенно [2]. Это ведет к более эффективному сотрудничеству команды [18].
- Интеграция позволяет автоматизировать ряд рутинных операций и тем самым упростить workflow разработки и управления. Например, больше не нужно вручную дублировать информацию — статусы и данные синхронизируются автоматически между инструментами [2]. Объединенная среда дает упрощенный рабочий процесс: меньше переключений между приложениями, единый поток задач [3]. Примеры — авто-триггеры Jira, переводящие задачи при событиях в репозитории [13]; или автоматическое логирование затраченного времени из Git-репозитория в систему учёта времени проекта [12]. Совместное использование инструментов ведёт к экономии времени и ресурсов, что в итоге означает снижение затрат проекта [16].
- При интегрированных системах значительно улучшается трассируемость (traceability) — возможность проследить связь между требованиями, задачами, исходным кодом, тестами и результатами [3][17]. Контроль версий и задач в связке позволяет точно знать состав сборок; проще проводить код-ревью, не выходя из контекста задачи, а также выявлять риски для выпуска заранее [13].
- Когда информация о ходе разработки мгновенно доступна менеджерам, они могут быстрее принимать решения и корректировать планы проекта [12]. Интеграция упрощает реакцию на изменения, синхронизируя бизнес-цели и техническую реализацию [12][20].

Приведенные преимущества подтверждаются практическими кейсами. Так, интеграция Jira и GitLab в одной из команд привела к улучшению совместной работы между менеджерами и разработчиками, повышению

прозрачности задач и автоматизации отслеживания проблем [18]. В другой ситуации внедрение комплексной системы управления проектами на предприятии позволило руководству постоянно быть в курсе статуса каждой задачи и ускорить запуск новых проектов [15][12]. Организации, внедрившие проактивные методы управления и современные инструменты, теряют в среднем в 28 раз меньше средств на неудачные инициативы [20].

### Проблемы и ограничения интеграции

- Переход к интегрированной системе требует изменений в процессах и культуре; сопротивление изменениям может стать барьером [7]. Сложность настройки интеграции часто требует участия системных аналитиков и администраторов.
- Классические PM-подходы и Agile различаются по циклам и метрикам; требуется согласование представлений [11]. Не всегда удаётся обеспечить полную двухстороннюю синхронизацию из-за различий моделей данных и ограничений API.
- Интегрированные платформы требуют инвестиций (лицензии/подписки, инфраструктура, обучение); важно оценивать окупаемость.
- Необходима программа обучения и постоянная поддержка (DevOps/администраторы); обновления сервисов могут нарушать интеграцию.
- Тотальная прозрачность может провоцировать микроменеджмент и перегрузку данными; важно настраивать уровень отчётности.

Ключевые трудности интеграции — прежде всего организационные. Без адаптации процессов, обучения и управленческой поддержки интеграция может не дать эффекта. Опыт показывает, что формальное внедрение без изменения методик приводит к дублированию информации и игнорированию системы пользователями [6]. Рекомендуется планировать интеграцию как проект организационных изменений с пилотами и поэтапным внедрением.

### Перспективы применения и направления дальнейших исследований

Рассмотренные в статье подходы к интеграции систем управления проектами и разработки ПО задают основу для последующих исследований. В дальнейшем

целесообразно разработать критерии выбора уровня интеграции (UI, обмен данными, API или событийное взаимодействие) в зависимости от зрелости организации и особенностей проектов. Перспективным направлением является создание универсальной модели интеграционного слоя, которая могла бы применяться в различных системах, а также разработка методов сквозной прослеживаемости задач от требований до релиза и алгоритмов автоматического выявления разрывов.

На практике рекомендуется начинать с пилотных решений — связки трекера задач и репозитория кода (например, Asana↔GitLab или Jira↔Git) с минимально необходимой автоматизацией статусов и проверок в CI/CD. Важно закрепить единый словарь артефактов (задача, эпик, релиз) и набор метрик (lead time, частота релизов, полнота связей задача↔коммит↔сборка), которые можно отслеживать в общей панели мониторинга. Такой подход позволит оценить эффекты интеграции в сопоставлении с управленческими практиками и обеспечить воспроизводимость экспериментов.

Для подтверждения результатов требуется экспериментальная проверка на реальных проектах: измерение изменений во времени выполнения задач, частоте релизов, числе дефектов и соблюдении сроков. Ожидается, что результаты таких исследований помогут обосновать новые методы и алгоритмы интеграции, включая модель выбора уровня интеграции, архитектуру интеграционного слоя и методику увязки показателей DevOps с показателями управления проектами.

### Заключение

Интеграция систем управления проектами с инструментами разработки программного обеспечения становится все более насущной потребностью. Она повышает прозрачность и управляемость, улучшает взаимодействие команды и снижает ошибки за счёт автоматизации [5]. Внедрение требует учёта организационных аспектов и выбора инструментов под масштаб предприятия. Правильно реализованная интеграция повышает гибкость управления и снижает риски провалов и перерасхода бюджета [20]. Концепции DevOps и BizDevOps расширяют интеграцию на весь жизненный цикл, и этот тренд будет усиливаться.

### ЛИТЕРАТУРА

1. Вакорин М.П., Симонов В.И., Данилов С.В. Оценка возможности использования среды разработки программного обеспечения GitLab в задачах управления проектами // Молодой ученый. — 2023. — № 11 (458). — С. 5–8.
2. DevOps Tec. [Asana + GitLab] Synchronized Integration of Project Management and Software Development // DevOps Tec Blog. — 2024. — July 18. — URL: <https://www.devops.com.my/post/asana-gitlab-integration-malaysia> (дата обращения: 10.09.2025).
3. Bokone A. How to integrate IT task management tools with version control systems? (FAQ) // Assembla Blog. — 2024. — Oct 27. — URL: <https://get.assembla.com/blog/5-it-task-management-faqs> (дата обращения: 12.09.2025).



4. Amazon Web Services. Что такое управление жизненным циклом приложений (ALM)? — URL: <https://aws.amazon.com/ru/what-is/application-lifecycle-management/> (дата обращения: 05.09.2025).
5. Microsoft. Обзор управления жизненным циклом приложений с Microsoft Power Platform — Microsoft Learn. — 12.02.2025. — URL: <https://learn.microsoft.com/ru-ru/power-platform/alm/overview-alm> (дата обращения: 05.09.2025).
6. Тулембаев А.Н., Адилова А.М., Серикбекулы А. Интеграция системы управления проектами на предприятии оборонно-промышленного комплекса Республики Казахстан // Экономика, предпринимательство и право. — 2020. — Т. 10, № 7. — С. 2029–2044.
7. Шаманин А.Ю., Трибелев А.А. Как интеграция проектного управления и системной инженерии улучшает результаты проектов // Труды XIV Всеросс. совещания по проблемам управления (ВСПУ-2024), Москва, 17–20 июня 2024 г. — М.: ИПУ РАН, 2024. — С. 1–5.
8. Мошаров А. Наводим порядок в управлении разработкой с помощью Gitlab и Jira // Хабр (блог компании «Северсталь»). — 2022. — 9 ноября. — URL: <https://habr.com/ru/companies/severstal/articles/698236/> (дата обращения: 08.09.2025).
9. Add-On Development. Method & Resources — Team Foundation Server allows integration of project management and development activities... — Company Website. — 2019. — URL: <https://aod.vn/method-resources.html> (дата обращения: 07.09.2025).
10. Интеграция управления программой и системной инженерии / под ред. Э. С. Ребентиша; пер. с англ. В. К. Батоврина и др. — М.: ДМК Пресс, 2020. — 584 с.
11. Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). — 6th ed. — Newtown Square, PA: PMI, 2017. — 592 p.
12. Teamhub. Project Management Tool Integration: Streamlining Your Workflow // Teamhub Blog. — 2024. — Jan 25. — URL: <https://teamhub.com/blog/project-management-tool-integration-streamlining-your-workflow/> (дата обращения: 10.09.2025).
13. Atlassian. Integrating with development tools — Jira Service Management Cloud Documentation. — 2021 (посл. обновл. 17 нояб. 2021). — URL: <https://confluence.atlassian.com/servicedeskcloud/integrating-with-development-tools-1097175801.html> (дата обращения: 08.09.2025).
14. Шантунова Ю. Top 7 Application Lifecycle Management (ALM) Tools // SaM Solutions Blog. — 2025. — Aug 11. — URL: <https://sam-solutions.com/blog/top-7-application-lifecycle-management-alm-tools/> (дата обращения: 06.09.2025).
15. Топ-10 лучших систем управления проектами в 2025 году // Блог Kickidler. — 2025. — 12 января. — URL: <https://www.kickidler.com/ru/info/top-10-luchshikh-sistem-upravleniya-proyektami> (дата обращения: 05.09.2025).
16. Cprime. What Is Jira Integration? (Blog) — Cprime.com, [б.р.]. — URL: <https://www.cprime.com/resources/blog/what-is-jira-integration/> (дата обращения: 08.09.2025).
17. Visure Solutions. Интеграция ALM и PLM: примеры, варианты использования — 2023. — URL: <https://visuresolutions.com/ru/гид-по-милостыне/интеграция-alm-и-plm/> (дата обращения: 10.09.2025).
18. Ružić J., Jovkić M. The Impact Of Software Tool Integration On Digital Project Management // Proc. 29th Int. Congress on Project Management (Belgrade, 2025). — 2025. — P. 45–52.
19. Boehm B., Turner R. Balancing Agility and Discipline: A Guide for the Perplexed. — Addison-Wesley, 2003. — 266 p.
20. Wimi Staff. 52 statistics on project management that you absolutely need to know // Wimi Blog. — 2022. — March 14. — URL: <https://www.wimi-teamwork.com/blog/52-statistics-on-project-management-that-you-absolutely-need-to-know/> (дата обращения 10.09.2025).

---

© Ильин Руслан Дмитриевич (mildgravity0@yandex.ru); Никишов Сергей Иванович (nikishov-si@ranepa.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»