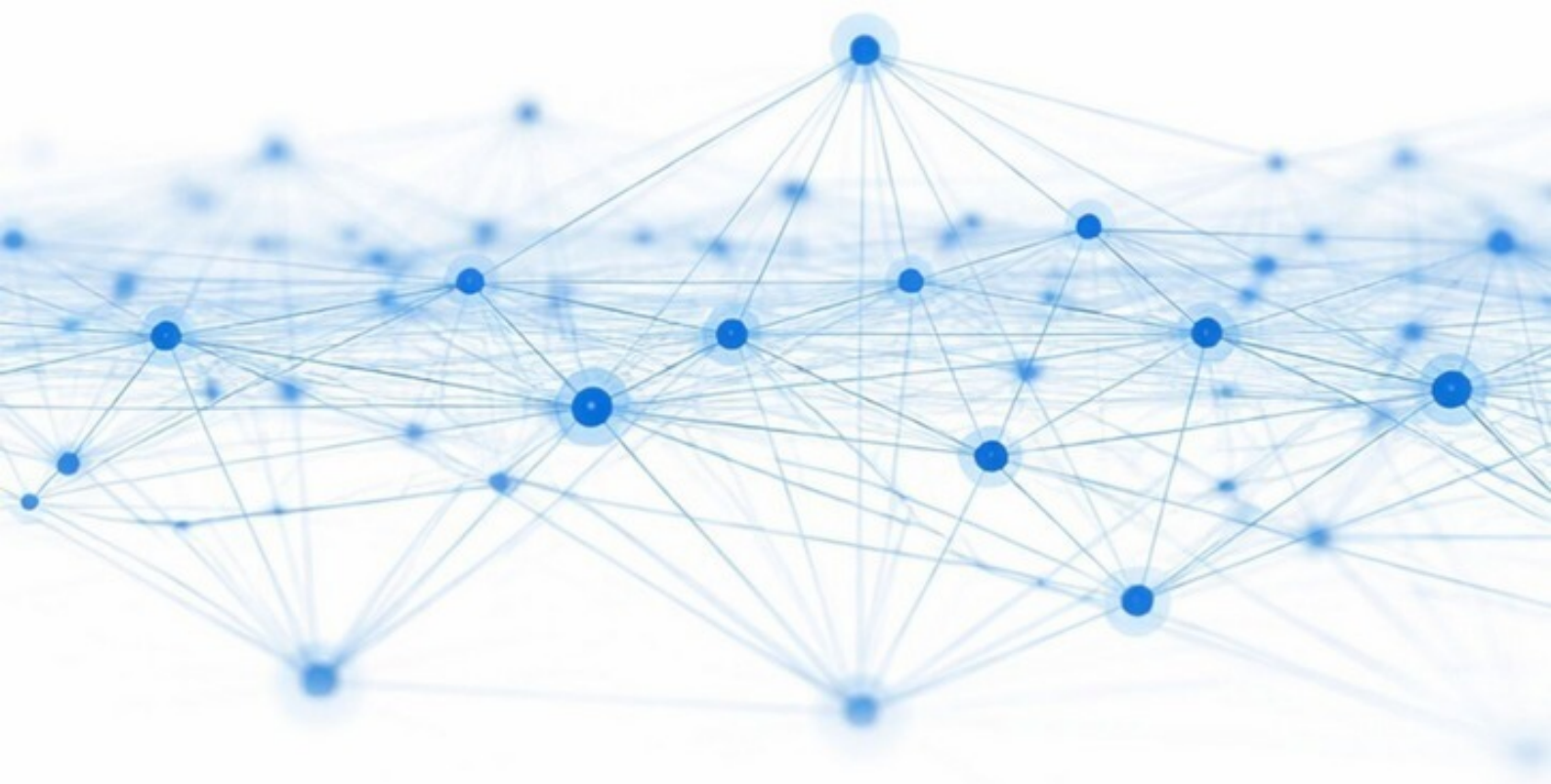


**ХИНКУ ВЛАДИСЛАВ**

---

**Экономика избыточного  
проектирования: количественная  
оценка отрицательной архитектурной  
ценности**

**МЕТОДИЧЕСКОЕ ПОСОБИЕ**



Владислав Хинку

**Экономика избыточного  
проектирования: количественная  
оценка отрицательной  
архитектурной ценности.  
Методическое пособие**

## **Хинку В.**

Экономика избыточного проектирования: количественная оценка отрицательной архитектурной ценности. Методическое пособие / В. Хинку —

Пособие посвящено оценке архитектурных решений программных систем с точки зрения эффективности и избыточной сложности. Представлены инструменты количественного анализа (CCI и POD), позволяющие выявлять неэффективные решения и обосновывать их оптимизацию. Рекомендуется разработчикам, архитекторам, руководителям IT-проектов и др. Рецензент: Арышева Галина Владиславовна — кандидат технических наук, доцент ТПУ, директор Бизнес-инкубатора социальных инноваций и социального предпринимательства ТГПУ.

# Содержание

Введение	6
1. Актуальность исследования	6
2. Цель и задачи исследования	8
3. Объект и предмет исследования	9
4. Научная новизна	10
5. Практическая значимость	11
Глава 1. Теоретические основы и обзор подходов к архитектуре программных систем	12
1.1. Архитектура программного обеспечения как объект экономического анализа	12
1.2. Понятие избыточного проектирования	14
1.3. Ненужная сложность как источник потерь	15
1.4. Технический долг и отрицательная архитектурная ценность	16
1.5. Влияние архитектуры на стоимость разработки и сопровождения	17
1.6. Обзор существующих подходов и исследований	18
1.7. Ограничения существующих моделей	20
Глава 2. Методологическая база и концепция оценки архитектурной эффективности	21
2.1. Источники данных и методика анализа	21
2.2. Подход к отбору и классификации кейсов	22
2.3. Понятие отрицательной архитектурной ценности	23
2.4. Общая модель оценки архитектурных решений	24
Конец ознакомительного фрагмента.	25

**Экономика избыточного  
проектирования: количественная оценка  
отрицательной архитектурной ценности  
Методическое пособие**

**Владислав Хинку**

© Владислав Хинку, 2026

ISBN 978-5-0069-9431-7

Создано в интеллектуальной издательской системе Ridero

## Введение

### 1. Актуальность исследования

В современных условиях цифровой трансформации архитектура программных систем рассматривается как один из ключевых факторов, определяющих эффективность разработки, устойчивость эксплуатации и экономические показатели технологических компаний. Традиционно архитектурные решения оцениваются с позиции создаваемой ценности – повышения масштабируемости, ускорения разработки, улучшения сопровождаемости и устойчивости систем. Однако практический опыт последних лет показывает, что значительная часть архитектурных решений приводит не к созданию, а к снижению совокупной эффективности за счет внедрения избыточной сложности.

Анализ эмпирических данных, включающий более 50 публичных кейсов и результаты опросов свыше 95 000 разработчиков, показывает, что избыточно спроектированные системы обходятся организациям в 2—8 раз дороже по сравнению с архитектурно обоснованными решениями, при медианном значении около 4,1 раза. При этом скорость разработки функциональности снижается на 40—60%, несмотря на использование современных технологических подходов.

Особенно ярко данная проблема проявляется при внедрении микросервисной архитектуры, облачных решений и распределенных систем без наличия соответствующих бизнес-требований. Так, в командах численностью менее 30 инженеров переход к микросервисам в 58% случаев приводит к снижению эффективности в первые два года эксплуатации. Преждевременная оптимизация инфраструктуры, в том числе при переходе к облачным моделям, может увеличивать затраты в среднем в 3,2 раза без существенного улучшения производственных характеристик системы.

Практические кейсы крупнейших технологических компаний подтверждают данные выводы. Консолидация более чем 140 микросервисов в единую систему позволила сократить затраты на инфраструктуру до 90% и ускорить разработку на 50%. Аналогичные результаты были достигнуты при отказе от избыточных распределенных решений и возврате к более простым архитектурным моделям. Эти примеры демонстрируют, что сложность архитектуры сама по себе не обладает ценностью и может выступать фактором прямых экономических потерь.

Дополнительные исследования показывают, что рост архитектурной сложности напрямую влияет на распределение времени инженерных команд. В высокосложных системах до 48% рабочего времени уходит на поддержку инфраструктуры, тогда как в рационально спроектированных системах данный показатель составляет около 12%. Это приводит к снижению продуктивности, увеличению числа инцидентов и ухудшению качества разработки.

Несмотря на масштаб проблемы, в существующих научных и прикладных исследованиях отсутствует единый количественный инструментарий, позволяющий оценивать экономические последствия архитектурных решений на ранних этапах проектирования. Большинство подходов сосредоточено на анализе технического долга, который предполагает отложенные издержки, тогда как проблема избыточного проектирования связана с формированием отрицательной ценности уже на стадии создания системы.

Актуальной задачей становится разработка методического подхода, позволяющего выявлять, количественно оценивать и предотвращать избыточную архитектурную сложность, а также обеспечивать соответствие архитектурных решений реальным бизнес-требованиям. Решение данной задачи имеет теоретическое значение для развития методов оценки про-

граммных систем и практическую значимость для повышения эффективности деятельности технологических компаний.

## 2. Цель и задачи исследования

Целью настоящего исследования является разработка методического подхода к выявлению, количественной оценке и предотвращению избыточного архитектурного проектирования в программных системах на основе анализа эмпирических данных и формирования прикладных инструментов оценки архитектурной сложности.

Для достижения поставленной цели решаются следующие задачи:

- провести анализ существующих подходов к оценке архитектуры программных систем и выявить их ограничения;
- исследовать влияние избыточной архитектурной сложности на стоимость разработки, скорость реализации функциональности и эксплуатационные характеристики систем;
- сформировать понятие отрицательной архитектурной ценности и определить ее ключевые признаки;
- разработать количественный инструмент оценки архитектурной сложности (индекс сложности и стоимости – ССИ);
- разработать метод выявления преждевременной оптимизации архитектурных решений (POD);
- провести эмпирический анализ архитектурных решений на основе реальных кейсов и статистических данных;
- определить закономерности влияния архитектурных решений на эффективность разработки и эксплуатации систем;
- разработать методику принятия архитектурных решений с учетом соотношения сложности и бизнес-требований;
- сформировать практические рекомендации по проектированию архитектуры программных систем.

### **3. Объект и предмет исследования**

Объектом исследования являются архитектурные решения в программных системах, применяемые при разработке и эксплуатации информационных продуктов в различных отраслях экономики.

Предметом исследования выступают методы и инструменты оценки архитектурной сложности, а также влияние избыточного проектирования на экономические и производственные показатели программных систем, включая стоимость разработки, скорость реализации функциональности и эксплуатационные затраты.

## 4. Научная новизна

Научная новизна исследования заключается в разработке методического подхода к количественной оценке избыточной архитектурной сложности программных систем и ее влияния на экономические и производственные показатели. В рамках работы сформировано и обосновано понятие отрицательной архитектурной ценности, отражающее ситуации, при которых архитектурные решения приводят к снижению эффективности системы уже на этапе ее создания за счет внедрения ненужной сложности.

Предложен индекс сложности и стоимости (ССИ), позволяющий количественно оценивать уровень архитектурной сложности с учетом факторов сервисной декомпозиции, инфраструктурного избытка, распределенности и инструментальной нагрузки. Разработан метод выявления преждевременной оптимизации архитектурных решений (POD), основанный на сопоставлении фактических характеристик системы с реальными требованиями к производительности, масштабируемости и надежности.

В ходе исследования установлены количественные зависимости между уровнем архитектурной сложности и ключевыми показателями эффективности, включая стоимость разработки, скорость реализации функциональности и загрузку инженерных команд. Предложена модель оценки соответствия архитектурных решений бизнес-требованиям, позволяющая выявлять избыточное проектирование на ранних этапах. Сформирована методика принятия архитектурных решений, основанная на ограничении уровня сложности и учете ресурсов команды разработки

## 5. Практическая значимость

Практическая значимость исследования заключается в возможности применения разработанного методического подхода и предложенных инструментов для повышения эффективности проектирования и эксплуатации программных систем. Разработанные в рамках работы решения могут использоваться при формировании архитектуры программных продуктов для оценки целесообразности применения различных архитектурных подходов и предотвращения избыточного проектирования. Полученные результаты представляют интерес для архитекторов и технических руководителей при принятии решений о внедрении микросервисной архитектуры, облачных решений и распределенных систем с учетом реальных бизнес-требований и ограничений.

Предложенные инструменты, включая индекс сложности и стоимости (ССИ) и метод выявления преждевременной оптимизации (POD), могут применяться при проведении архитектурного аудита существующих систем с целью выявления избыточной сложности, оптимизации архитектурных решений и снижения эксплуатационных затрат. Кроме того, результаты исследования могут использоваться при планировании развития программных продуктов с учетом доступных ресурсов и возможностей команды разработки, что позволяет повысить обоснованность принимаемых решений и снизить риски неэффективных архитектурных изменений.

Практическое применение разработанного подхода обеспечивает снижение затрат на разработку и сопровождение программных систем, повышение скорости реализации функциональности и улучшение управляемости архитектуры. Материалы исследования также могут быть использованы в образовательном процессе при подготовке специалистов в области разработки программного обеспечения и архитектуры информационных систем.

# **Глава 1. Теоретические основы и обзор подходов к архитектуре программных систем**

## **1.1. Архитектура программного обеспечения как объект экономического анализа**

Архитектура программного обеспечения традиционно рассматривается как совокупность принципов, решений и структур, определяющих организацию системы, способы взаимодействия ее компонентов и выбор используемых технологий. В инженерной практике архитектура выполняет роль основы, на которой строится весь жизненный цикл программного продукта, включая разработку, внедрение, сопровождение и развитие. При этом в последние годы наблюдается смещение акцента с исключительно технического понимания архитектуры к ее рассмотрению как фактора, непосредственно влияющего на экономические результаты деятельности организаций.

С экономической точки зрения архитектура программной системы определяет структуру затрат на всех этапах жизненного цикла. Выбор архитектурных решений влияет на объем ресурсов, необходимых для разработки функциональности, сложность сопровождения системы, скорость внедрения изменений и устойчивость к эксплуатационным нагрузкам. Ошибки, допущенные на этапе проектирования архитектуры, обладают кумулятивным эффектом и приводят к многократному увеличению затрат в дальнейшем, что делает архитектуру одним из наиболее значимых факторов формирования совокупной стоимости владения программным продуктом.

Современные подходы к разработке программных систем зачастую ориентированы на внедрение сложных архитектурных решений, включая микросервисные структуры, распределенные системы и облачные инфраструктуры. Предполагается, что такие решения обеспечивают гибкость, масштабируемость и устойчивость систем. Однако эмпирические данные показывают, что использование сложных архитектурных подходов без наличия соответствующих требований может приводить к обратному эффекту – росту затрат, снижению скорости разработки и увеличению эксплуатационных рисков.

В рамках экономического анализа архитектуры особое значение приобретает сопоставление затрат и создаваемой ценности. Архитектурные решения должны оцениваться не только по их техническим характеристикам, но и по их способности обеспечивать эффективное использование ресурсов и достижение бизнес-целей. При этом сложность архитектуры выступает одним из ключевых факторов, определяющих соотношение затрат и результата. Увеличение сложности системы без объективной необходимости приводит к росту операционных издержек, увеличению времени разработки и снижению управляемости системы.

Дополнительным фактором является влияние архитектуры на организацию работы инженерных команд. Более сложные архитектурные решения требуют увеличения координации, усложняют процессы разработки и повышают когнитивную нагрузку на специалистов. Это приводит к перераспределению времени в сторону обслуживания инфраструктуры и снижению доли времени, направленного на создание бизнес-ценности. В результате архитектура становится не только техническим, но и организационно-экономическим фактором, влияющим на эффективность работы команды.

В условиях роста масштабов и сложности программных систем возникает необходимость перехода от качественной оценки архитектурных решений к количественным методам анализа. Это предполагает разработку инструментов, позволяющих измерять уровень архитектурной

сложности, оценивать ее влияние на ключевые показатели эффективности и выявлять случаи, в которых архитектурные решения приводят к снижению общей эффективности системы. Отсутствие таких инструментов затрудняет принятие обоснованных решений и повышает риск внедрения избыточной сложности.

Рассмотрение архитектуры программного обеспечения как объекта экономического анализа позволяет сформировать основу для разработки методических подходов, направленных на оптимизацию архитектурных решений, снижение затрат и повышение эффективности разработки и эксплуатации программных систем.

## 1.2. Понятие избыточного проектирования

В современной практике разработки программных систем под избыточным проектированием понимается создание архитектурных решений, уровень сложности которых превышает фактические требования к системе. Речь идет не просто о сложных системах, а о тех случаях, когда сложность не обусловлена реальными нагрузками, масштабом или бизнес-задачами, а формируется как следствие неверных предпосылок на этапе проектирования.

Распространение данного явления связано с активным внедрением современных технологических подходов, включая микросервисную архитектуру, облачные решения и распределенные системы. Эти подходы часто воспринимаются как универсальные и применяются вне зависимости от контекста. В результате архитектура формируется с расчетом на потенциальные сценарии, которые в действительности не реализуются.

Избыточное проектирование может проявляться в различных формах. На практике это выражается в чрезмерной декомпозиции системы на большое количество сервисов, внедрении сложной инфраструктуры с избыточным резервированием, использовании распределенных механизмов обработки данных при отсутствии требований к минимальной задержке, а также в добавлении дополнительных уровней абстракции. Подобные решения усложняют систему, но не обеспечивают прироста полезного результата.

С экономической точки зрения подобная архитектура формирует дополнительные затраты, не обеспеченные соответствующей ценностью. Эмпирические данные показывают, что избыточно спроектированные системы требуют значительно больше ресурсов как на этапе разработки, так и в процессе эксплуатации. Увеличивается время реализации функциональности, возрастает нагрузка на инфраструктуру и усложняется сопровождение системы.

Отдельного внимания заслуживает связь избыточного проектирования с преждевременной оптимизацией. Архитектурные решения принимаются исходя из гипотетических требований, которые не подтверждены фактическими данными о нагрузке или использовании системы. В результате создается архитектура, рассчитанная на значительно более высокий уровень требований, чем тот, который действительно необходим.

В отличие от технического долга, предполагающего отложенные издержки, избыточное проектирование приводит к негативному эффекту уже на этапе создания системы. Архитектура становится источником дополнительных затрат без соответствующего увеличения полезного результата. Это позволяет рассматривать избыточное проектирование как самостоятельное явление, требующее отдельного анализа.

Понимание природы избыточного проектирования необходимо для формирования практических подходов к его выявлению и предотвращению. Это требует перехода к количественным методам анализа, позволяющим сопоставлять уровень архитектурной сложности с реальными требованиями и ресурсами системы.

### 1.3. Ненужная сложность как источник потерь

Ненужная сложность в архитектуре программных систем представляет собой совокупность элементов, решений и взаимосвязей, которые не обусловлены реальными требованиями к системе, но при этом оказывают влияние на ее функционирование, разработку и сопровождение. В отличие от необходимой сложности, связанной с решением объективно сложных задач, ненужная сложность возникает в результате избыточных архитектурных решений и не приносит дополнительной ценности.

Формирование ненужной сложности происходит на этапе проектирования, когда в систему закладываются архитектурные механизмы, ориентированные на потенциальные, но не подтвержденные сценарии. Это может выражаться в избыточной декомпозиции, внедрении дополнительных слоев абстракции, использовании сложных схем взаимодействия компонентов и избыточной инфраструктурной организации. В результате система становится более трудной для понимания, разработки и поддержки.

С экономической точки зрения ненужная сложность напрямую связана с ростом затрат. Увеличивается объем работ на этапе разработки, поскольку требуется реализовывать и поддерживать большее количество компонентов. Усложняется процесс внедрения изменений, что приводит к увеличению времени разработки функциональности. Дополнительно возрастает нагрузка на инфраструктуру, что отражается на эксплуатационных расходах.

Влияние ненужной сложности проявляется и в изменении структуры затрат внутри команды разработки. Значительная часть времени начинает расходоваться не на создание новой функциональности, а на обслуживание архитектуры, координацию между компонентами и устранение возникающих проблем. Эмпирические данные показывают, что в системах с высокой архитектурной сложностью доля времени, затрачиваемого на инфраструктуру, может достигать 48%, тогда как в рационально спроектированных системах этот показатель составляет около 12%.

Дополнительным последствием является рост числа ошибок и инцидентов. Увеличение количества взаимодействующих компонентов повышает вероятность возникновения сбоев, а сложность системы затрудняет их диагностику и устранение. Это снижает общую надежность системы и увеличивает затраты на ее сопровождение.

Ненужная сложность может рассматриваться как самостоятельный источник экономических потерь, проявляющийся на всех этапах жизненного цикла программной системы. Ее наличие снижает эффективность использования ресурсов, замедляет развитие продукта и усложняет управление архитектурой. Это требует разработки инструментов для ее выявления и количественной оценки.

## 1.4. Технический долг и отрицательная архитектурная ценность

Понятие технического долга широко используется в программной инженерии для описания последствий компромиссных решений, принимаемых с целью ускорения разработки. В классическом понимании технический долг возникает в ситуациях, когда разработчики осознанно упрощают архитектуру или откладывают часть работ, что позволяет быстрее реализовать функциональность, но приводит к необходимости последующего рефакторинга и дополнительным затратам в будущем.

Технический долг предполагает, что изначально созданная система обладает определенной ценностью и выполняет поставленные задачи, однако требует доработки и оптимизации. Затраты, связанные с техническим долгом, носят отложенный характер и проявляются по мере развития системы. При этом сам факт наличия долга не является исключительно негативным явлением, поскольку в ряде случаев он может быть оправдан с точки зрения скорости вывода продукта на рынок.

В отличие от технического долга, отрицательная архитектурная ценность характеризует ситуации, при которых архитектурные решения приводят к снижению эффективности системы уже на этапе ее создания. Речь идет о случаях, когда в архитектуру изначально закладывается избыточная сложность, не обусловленная требованиями, что формирует дополнительные затраты без соответствующей полезной отдачи.

Ключевое различие между рассматриваемыми понятиями заключается в характере влияния на экономические показатели системы. Технический долг создает обязательства, которые необходимо погасить в будущем, тогда как отрицательная архитектурная ценность формирует прямые потери с момента внедрения архитектурного решения. В первом случае система сначала приносит ценность, а затем требует дополнительных вложений, во втором – затраты возникают без эквивалентного результата.

Отрицательная архитектурная ценность проявляется в тех же формах, что и избыточное проектирование: чрезмерная декомпозиция, избыточная инфраструктура, сложные схемы взаимодействия компонентов и внедрение технологий, не соответствующих масштабу задачи. Однако в данном случае акцент смещается с описания архитектурных решений на оценку их влияния на экономические показатели.

Эмпирические данные показывают, что системы с высоким уровнем архитектурной сложности демонстрируют значительное снижение эффективности. Увеличиваются затраты на разработку, снижается скорость реализации функциональности, возрастает доля времени, затрачиваемого на инфраструктуру, и ухудшается управляемость системы. Эти эффекты проявляются сразу после внедрения архитектуры и не требуют длительного периода накопления, что отличает их от последствий технического долга.

Разграничение технического долга и отрицательной архитектурной ценности имеет важное практическое значение. Оно позволяет корректно оценивать причины снижения эффективности системы и выбирать соответствующие методы управления. В случае технического долга приоритетом становится рефакторинг и улучшение качества кода, тогда как при наличии отрицательной архитектурной ценности требуется пересмотр архитектурных решений и снижение уровня сложности системы.

Понимание различий между этими явлениями создает основу для разработки количественных инструментов оценки архитектуры, направленных на выявление случаев, в которых архитектурные решения приводят к прямым экономическим потерям.

## 1.5. Влияние архитектуры на стоимость разработки и сопровождения

Архитектура программной системы оказывает прямое влияние на формирование затрат на всех этапах ее жизненного цикла. На стадии разработки архитектурные решения определяют структуру системы, количество компонентов, характер их взаимодействия и используемые технологии, что в совокупности формирует объем необходимых ресурсов. Выбор архитектуры влияет не только на начальные затраты, но и на динамику их изменения по мере развития продукта.

Одним из ключевых факторов является сложность архитектурной структуры. Увеличение количества компонентов и связей между ними приводит к росту трудоемкости разработки, поскольку возрастает объем кода, который необходимо реализовать и поддерживать. Дополнительно усложняется процесс тестирования, так как требуется учитывать большее количество сценариев взаимодействия элементов системы.

Влияние архитектуры особенно заметно при реализации изменений. В системах с высокой связностью и сложной структурой внедрение новой функциональности требует координации между несколькими компонентами, что увеличивает время разработки и повышает вероятность возникновения ошибок. Напротив, в рационально спроектированных системах изменения могут вноситься локально, без существенного воздействия на другие части системы.

Существенную роль играет влияние архитектуры на эксплуатационные затраты. Использование сложных инфраструктурных решений, включая распределенные системы и облачные сервисы с избыточным резервированием, приводит к увеличению расходов на поддержку системы. Эмпирические данные показывают, что избыточно спроектированные архитектуры могут увеличивать совокупные затраты в несколько раз по сравнению с более простыми решениями.

Дополнительным фактором является влияние архитектуры на скорость разработки. В системах с высокой архитектурной сложностью наблюдается значительное снижение темпов реализации функциональности. Это связано с необходимостью согласования изменений, увеличением времени на развертывание и сложностью диагностики проблем. В ряде случаев снижение скорости разработки достигает 40—60%, что оказывает существенное влияние на конкурентоспособность продукта.

Архитектура также определяет структуру затрат на сопровождение системы. Увеличение сложности приводит к росту затрат на поддержку, поскольку требуется больше ресурсов для мониторинга, устранения ошибок и обеспечения стабильной работы. Дополнительно возрастает зависимость от квалификации специалистов, что повышает стоимость команды разработки.

Влияние архитектуры распространяется и на организационные процессы. Сложные архитектурные решения требуют более высокого уровня координации внутри команды, что увеличивает накладные расходы на управление и коммуникацию. Это приводит к снижению общей эффективности работы и увеличению времени, необходимого для реализации задач.

Рассмотрение архитектуры как фактора, определяющего структуру затрат, позволяет перейти от интуитивного выбора решений к их обоснованной оценке. Это создает предпосылки для разработки инструментов, позволяющих количественно анализировать влияние архитектурных решений на стоимость разработки и сопровождения программных систем.

## 1.6. Обзор существующих подходов и исследований

Вопросы оценки архитектурных решений и их влияния на эффективность программных систем рассматриваются в рамках нескольких направлений исследований, включая теорию технического долга, анализ сложности программных систем, а также исследования в области микросервисной архитектуры и облачных вычислений. Каждое из этих направлений вносит вклад в понимание архитектурных процессов, однако не формирует целостного инструментария для количественной оценки избыточной сложности.

Одним из наиболее распространенных подходов является концепция технического долга, которая описывает последствия компромиссных решений в разработке. В рамках данного подхода внимание сосредоточено на отложенных издержках, возникающих в результате упрощения архитектуры или откладывания работ. Несмотря на широкое распространение, данный подход не охватывает ситуации, при которых архитектура изначально формирует избыточные затраты без создания дополнительной ценности.

Отдельное направление исследований связано с анализом сложности программных систем. В ряде работ подчеркивается различие между необходимой и случайной сложностью, однако такие исследования, как правило, носят качественный характер и не предлагают количественных методов оценки экономических последствий. Это ограничивает возможность практического применения данных подходов при принятии архитектурных решений.

Существенное внимание уделяется исследованиям микросервисной архитектуры. В научной и прикладной литературе подчеркиваются преимущества данного подхода, включая независимость компонентов, гибкость масштабирования и возможность использования различных технологий. Вместе с тем отдельные работы указывают на необходимость соблюдения определенных условий для успешного применения микросервисов, таких как достаточный размер команды и зрелость процессов разработки. Отсутствие количественных критериев делает такие рекомендации трудно применимыми на практике.

Исследования в области облачных вычислений традиционно акцентируют внимание на преимуществах эластичности и модели оплаты по факту использования ресурсов. Однако эмпирические данные показывают, что при предсказуемых нагрузках использование облачной инфраструктуры может приводить к увеличению затрат по сравнению с альтернативными решениями. Это указывает на необходимость более детального анализа экономической целесообразности архитектурных решений.

Отдельное направление представлено подходами, ориентированными на снижение технологической сложности, в которых предлагается ограничивать использование новых технологий и отдавать предпочтение более простым решениям. Данные подходы формируют важные практические рекомендации, однако не обеспечивают количественной оценки последствий архитектурных решений и не позволяют формализовать процесс их выбора.

Современные эмпирические исследования, основанные на анализе практических кейсов и данных опросов разработчиков, демонстрируют наличие устойчивых закономерностей между уровнем архитектурной сложности и ключевыми показателями эффективности. В частности, наблюдается рост затрат, снижение скорости разработки и увеличение доли времени, затрачиваемого на инфраструктуру, при увеличении сложности архитектуры. Несмотря на это, в большинстве работ отсутствует системный подход к количественной оценке данных эффектов.

Анализ существующих исследований показывает, что на сегодняшний день отсутствует единый методический подход, позволяющий количественно оценивать избыточную архитектурную сложность и ее влияние на экономические показатели. Это определяет необходимость

разработки инструментов, обеспечивающих сопоставление архитектурных решений с реальными требованиями и ресурсами системы.

## 1.7. Ограничения существующих моделей

Несмотря на значительное количество исследований, посвященных архитектуре программных систем, существующие подходы обладают рядом ограничений, которые снижают их применимость для практической оценки архитектурных решений. Основная проблема заключается в отсутствии количественного инструментария, позволяющего сопоставлять уровень архитектурной сложности с экономическими результатами.

Подходы, основанные на анализе технического долга, ориентированы преимущественно на оценку отложенных издержек и не учитывают ситуации, в которых архитектурные решения изначально формируют избыточные затраты. В результате такие модели не позволяют выявлять случаи отрицательной архитектурной ценности на ранних этапах проектирования.

Методы оценки сложности программных систем, как правило, сосредоточены на структурных характеристиках кода или архитектуры, но не связывают их с экономическими показателями. Отсутствие прямой связи между сложностью и затратами ограничивает возможность использования этих методов при принятии управленческих решений.

Подходы, рассматривающие микросервисную архитектуру и облачные решения, чаще всего носят рекомендательный характер. В них формулируются общие принципы и условия применения, однако отсутствуют четкие количественные критерии, позволяющие определить границы целесообразности использования данных технологий. Это приводит к тому, что архитектурные решения принимаются на основе предположений, а не объективных показателей.

Дополнительным ограничением является недостаточное внимание к эмпирическим данным. Несмотря на наличие отдельных исследований, демонстрирующих влияние архитектурной сложности на эффективность разработки и эксплуатации, в большинстве случаев отсутствует систематизация полученных результатов и их интеграция в единый аналитический подход.

Существующие модели также не учитывают влияние архитектурных решений на организационные процессы, включая распределение времени внутри команды, когнитивную нагрузку и сложность координации. Это ограничивает полноту оценки и не позволяет учитывать все факторы, влияющие на эффективность разработки.

Отсутствие комплексного подхода, объединяющего технические, экономические и организационные аспекты архитектуры, приводит к тому, что процесс выбора архитектурных решений остается во многом интуитивным. Это повышает риск внедрения избыточной сложности и формирования дополнительных затрат.

Выявленные ограничения определяют необходимость разработки методического подхода, позволяющего количественно оценивать архитектурные решения, выявлять случаи избыточного проектирования и обеспечивать их соответствие реальным требованиям системы. Такой подход должен учитывать не только технические характеристики, но и их влияние на экономические и организационные показатели.

## **Глава 2. Методологическая база и концепция оценки архитектурной эффективности**

### **2.1. Источники данных и методика анализа**

В основе проведенного исследования лежит комбинированный подход, включающий анализ эмпирических данных, полученных из публичных кейсов, отраслевых опросов и практического опыта разработки программных систем. Такой подход позволяет обеспечить комплексное рассмотрение архитектурных решений с учетом как количественных, так и качественных факторов.

Эмпирическая база исследования включает анализ более 50 публичных кейсов, связанных с изменением архитектуры программных систем, в том числе переходом от сложных архитектурных решений к более простым моделям. В рамках анализа рассматривались такие параметры, как количество компонентов системы, структура архитектуры, затраты на инфраструктуру, скорость разработки и результаты изменений архитектуры.

Дополнительно использованы данные отраслевых опросов, охватывающих более 95 000 разработчиков и организаций. Эти данные позволяют оценить распространенные практики применения архитектурных подходов, выявить типовые проблемы и определить влияние архитектурной сложности на ключевые показатели эффективности разработки. В частности, анализировались показатели удовлетворенности разработчиков, распределение времени между разработкой и поддержкой инфраструктуры, а также частота использования различных архитектурных решений.

В исследовании также учитывались практические кейсы разработки программных систем, охватывающие различные отрасли, включая коммерческие и технологические проекты. Это позволило дополнить статистические данные наблюдениями, отражающими реальные условия применения архитектурных решений.

Методика анализа основана на сопоставлении архитектурных характеристик систем с их экономическими и производственными показателями. В рамках исследования рассматривалась взаимосвязь между уровнем архитектурной сложности, затратами на разработку и эксплуатацию, а также скоростью реализации функциональности. Для выявления закономерностей использовался сравнительный анализ различных архитектурных подходов, а также оценка изменений показателей до и после трансформации архитектуры.

Отдельное внимание уделено выявлению факторов, формирующих избыточную сложность. Для этого анализировались такие параметры, как количество компонентов системы, уровень распределенности, объем используемой инфраструктуры и разнообразие применяемых технологий. Сопоставление этих факторов с фактическими требованиями к системе позволило выявить случаи, в которых архитектурные решения не соответствуют реальным потребностям.

Применение комбинированной методики, основанной на анализе эмпирических данных и практических кейсов, обеспечивает возможность выявления устойчивых закономерностей и формирует основу для разработки количественных инструментов оценки архитектурной сложности.

## 2.2. Подход к отбору и классификации кейсов

Отбор кейсов для анализа осуществлялся с учетом необходимости получения репрезентативной выборки архитектурных решений, отражающих различные сценарии применения и трансформации программных систем. В исследование включались случаи, в которых наблюдалось существенное изменение архитектуры, сопровождающееся измеримыми изменениями экономических и производственных показателей.

Ключевым критерием отбора являлось наличие достоверных данных, позволяющих сопоставить состояние системы до и после изменения архитектуры. При этом учитывались такие параметры, как количество компонентов системы, структура архитектуры, уровень распределенности, затраты на инфраструктуру, скорость разработки и эксплуатационные характеристики. Предпочтение отдавалось кейсам, в которых изменения архитектуры сопровождались количественно выраженными результатами.

В рамках исследования рассматривались несколько типов кейсов. К первой группе относятся случаи отказа от избыточно сложных архитектурных решений, включая консолидацию микросервисных систем и упрощение распределенных архитектур. Ко второй группе отнесены кейсы, связанные с оптимизацией инфраструктуры, в том числе пересмотром облачных решений и сокращением избыточного резервирования ресурсов. Третью группу составляют примеры внедрения сложных архитектурных подходов без достаточных оснований, приводящие к увеличению затрат и снижению эффективности.

Классификация кейсов осуществлялась на основе уровня архитектурной сложности и характера влияния на показатели системы. Выделялись три основные категории: избыточно спроектированные системы, характеризующиеся высоким уровнем сложности и отрицательными экономическими результатами; рационально спроектированные системы, обеспечивающие баланс между сложностью и требованиями; а также системы с недостаточным уровнем архитектурной проработки, в которых наблюдается накопление технического долга.

Дополнительно применялся подход, основанный на сопоставлении архитектурных характеристик с требованиями к системе. Это позволяло выявлять случаи, в которых архитектурные решения существенно превышают фактические потребности по таким параметрам, как производительность, масштабируемость и доступность. Подобные расхождения рассматривались как признаки избыточного проектирования.

Анализ кейсов проводился с использованием сравнительного метода, предполагающего сопоставление различных архитектурных решений в сходных условиях. Это позволило выявить устойчивые зависимости между уровнем архитектурной сложности и ключевыми показателями эффективности, включая затраты, скорость разработки и нагрузку на команду.

Применяемый подход к отбору и классификации кейсов обеспечивает системность анализа и создает основу для последующего формирования количественных моделей оценки архитектурной сложности.

## 2.3. Понятие отрицательной архитектурной ценности

В рамках настоящего исследования вводится понятие отрицательной архитектурной ценности, отражающее ситуации, при которых архитектурные решения приводят к снижению эффективности программной системы и формированию дополнительных затрат без соответствующего увеличения полезного результата. Данное понятие позволяет рассматривать архитектуру не только как источник потенциальной ценности, но и как фактор, способный генерировать экономические потери.

Отрицательная архитектурная ценность возникает в случаях, когда уровень сложности архитектуры превышает фактические требования к системе. Это проявляется в несоответствии между реализованными возможностями и реальными потребностями, что приводит к избыточному использованию ресурсов. В таких условиях архитектура перестает выполнять функцию оптимизации процессов разработки и эксплуатации и становится источником дополнительных издержек.

Ключевой особенностью отрицательной архитектурной ценности является ее проявление на ранних этапах жизненного цикла системы. В отличие от технического долга, который формируется и накапливается со временем, отрицательная ценность возникает непосредственно в момент внедрения архитектурного решения. Это означает, что система изначально создается с избыточной сложностью, которая не обеспечивает дополнительной эффективности.

Проявления отрицательной архитектурной ценности включают чрезмерную декомпозицию системы, избыточную инфраструктурную организацию, использование распределенных решений при отсутствии соответствующих требований, а также внедрение дополнительных уровней абстракции. Все перечисленные факторы увеличивают сложность системы, не оказывая положительного влияния на ее функциональные характеристики.

Эмпирические данные подтверждают наличие устойчивой зависимости между уровнем архитектурной сложности и снижением эффективности разработки и эксплуатации. Системы с высоким уровнем сложности демонстрируют увеличение затрат, снижение скорости реализации функциональности и рост нагрузки на инженерные команды. Эти эффекты носят системный характер и проявляются независимо от конкретной предметной области.

С экономической точки зрения отрицательная архитектурная ценность может рассматриваться как результат несоответствия между затратами на реализацию архитектурных решений и получаемыми результатами. В таких случаях увеличение затрат не сопровождается ростом эффективности, что приводит к снижению общей производительности системы.

Введение понятия отрицательной архитектурной ценности создает основу для разработки количественных методов оценки архитектурных решений. Это позволяет перейти от качественного описания проблем к формированию инструментов, обеспечивающих выявление и предотвращение избыточного проектирования.

## 2.4. Общая модель оценки архитектурных решений

Оценка архитектурных решений в программных системах требует перехода от качественного описания к формализованному анализу, учитывающему взаимосвязь между архитектурной сложностью, затратами и достигаемыми результатами. В рамках настоящего исследования предлагается рассматривать архитектуру как систему, характеризующуюся совокупностью параметров, каждый из которых оказывает влияние на экономические и производственные показатели.

В основе модели лежит сопоставление трех ключевых компонентов: требований к системе, реализованных архитектурных характеристик и фактических затрат. Требования определяют необходимый уровень производительности, масштабируемости, доступности и функциональности. Архитектурные решения формируют набор реализованных возможностей системы, включая структуру компонентов, уровень распределенности и используемую инфраструктуру. Затраты отражают объем ресурсов, необходимых для разработки, внедрения и эксплуатации системы.

Несоответствие между указанными компонентами является источником снижения эффективности. В случае, если архитектурные характеристики существенно превышают требования, формируется избыточная сложность, приводящая к росту затрат без увеличения полезного результата. Если же архитектура не соответствует требованиям, возникают ограничения, влияющие на производительность и устойчивость системы. Баланс между требованиями и архитектурными решениями является основным условием эффективного проектирования.

В рамках предложенной модели архитектурная сложность рассматривается как количественный параметр, подлежащий измерению. Это позволяет оценивать степень отклонения архитектурных решений от оптимального уровня и выявлять случаи избыточного проектирования. Ключевым элементом является возможность сопоставления уровня сложности с экономическими показателями, что обеспечивает практическую применимость модели.

Модель также учитывает влияние архитектуры на организационные процессы. Увеличение сложности приводит к росту затрат времени на координацию, усложняет процессы разработки и повышает нагрузку на специалистов. Это отражается на скорости реализации функциональности и общей эффективности работы команды.

## **Конец ознакомительного фрагмента.**

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.