

Saint HighLoad++ 2026 · 22 июня

State of AI4SDLC: AI меняет разработку

Индустрия, крупный финтех на 10 000+ инженеров
и реальная работа инженера



**Saint
HighLoad++**
2026

Saint
in

План доклада

Навигация

Маршрут

- Индустрия: внедрение и agent-based SDLC.
- План: AI-платформа, IDP, agent surface.
- Что есть: gateways, context, agent mode.
- Как мерим эффект и работу инженеров.

Часть 1

01

Что происходит с индустрией

Внедрение уже случился: исследования сходятся в
оценках. А вот эффект приходит, только когда меняются
процессы



**Saint
HighLoad++**
2026

AI перестал быть экспериментом в IDE

Теперь нужно управлять новой физикой разработки

Контекст

Shift

Массовый внедрение

AI уже пишет код, тесты, docs и объяснения.

Неравный эффект

Кодинг ускоряется раньше поставки изменений.

Новый вопрос

Управлять внедрением, пропускной способностью разработки, quality/risk и economics.

Откуда мы знаем, что происходит

Исследования

Данные

Индустриальные отчеты и AI4SDLC дают один сигнал

Индустриальные исследования

- DORA и AI-assisted development reports.
- Кейсы: Google, Microsoft, GitHub, Meta, Uber.

Исследование AI4SDLC

- 50+ исследований 2023–2025.
- Опрос инженеров и техлидов в России.

Внедрение высокий, доверие ниже

Сигналы повторяются в исследованиях

Сигналы

Signals

Повседневный инструмент

AI стал частью рабочего дня.

Эффект неравномерен

Эффект зависит от задач и зрелости процессов.

Поставка отстаёт

Нужны review, tests и evals.

Кодинг быстрее, поставка не ускори́лась

Главный тезис

Bottleneck

Старые узкое местоs растут

- Review, тесты, релизы получают изменения.
- Слабая постановка быстрее дает неверный код.
- Без метрик команда спорит об ощущениях.
- Узкое место — постановка и проверка.

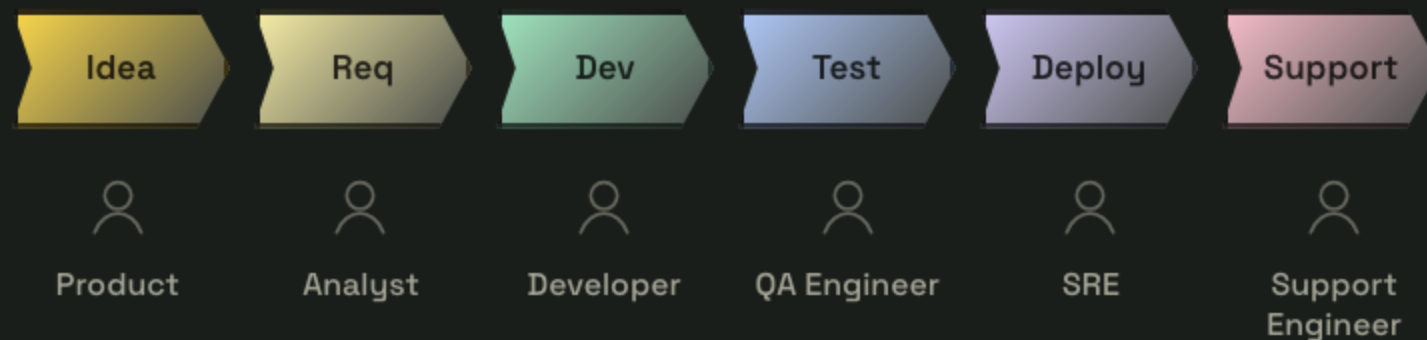
Role-based SDLC и agent-based SDLC

В SE 2.0 человек меньше пишет строки и больше управляет намерением, контекстом и проверкой

Сдвиг процессов

SE 1.0 → 2.0

Software Engineering 1.0 (Role-based SDLC)



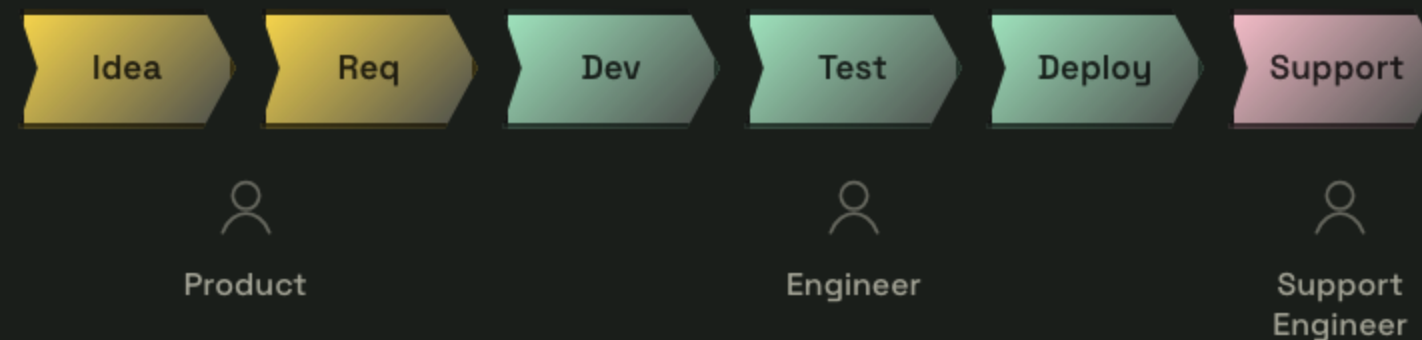
Потери на передачах работы

AI-сценарии внутри ролей

Локальные оптимизации

На brownfield-проектах обкатываем AI-сценарии по ролям

Software Engineering 2.0 (Agent-based SDLC)



Меньше потерь на передачах работы

Быстрее e2e-сценарии

На greenfield-проектах пробуем agent-based-разработку

Забираем наработки по сценариям

Забираем наработки по сценариям

Спесс вернулись: интерфейс для агента

Human-agent contract

Спецификации

SDD

- Агенту нужны intent, границы, acceptance criteria.
- Инструменты есть: Спес Kit, OpenСпес, SDD.
- Спецификация связана с задачами, тестами, review.
- Плохая спецификация масштабирует ошибку.

Часть 2

02

План изменений в финтехе

AI-платформа, agent-first IDP, агентная поверхность и
модель угроз на масштабе 10 000+ инженеров



**Saint
HighLoad++**
2026

Масштаб задает требования к решению

На 10 000+ инженеров нужна платформа

Контекст

Scale

Масштаб

Тысячи сервисов, репозиториев и стеков.

Безопасность

Данные, секреты и PII требуют свой контур контроля.

Экономика

Token/infra cost считается на уровне компании.

North Star программы: баланс трех осей

Метрики

North Star

Базовая линия, защитные ограничения и владелец у каждой оси

Скорость

Lead/cycle time: от задачи до prod.

Качество

Defects, incidents, rollback, security findings.

Экономика

Tokens, infra и человеко-время vs эффект.

IDP ЖИВ: GUI уходит

Контроль остается в платформе

Agent-first IDP

GUI → agents

В агент

- Intent и следующее действие.
- IDE, wiki, CI/CD, observability.
- Natural language к capabilities.

В платформе

- Доменная логика, policy, state changes.
- Лимиты, квоты, cost, blast radius.
- GUI: настройки, расследования, approvals.

Статья: [↗ tellmeabout.tech](https://tellmeabout.tech)

Три слоя agent-first платформы

Agent-first IDP

Layers

Model gateway, tool/MCP gateway и capability registry решают разные задачи и не заменяют друг друга

01 Модельный шлюз

AI / MODEL GATEWAY

Слой между вашим кодом и LLM-провайдерами.
Не каталог tools и не registry.

Apps / agents

LLM-запросы



Gateway

auth · quotas · cache ·
cost · PII



LLM API

model providers

OPEN SOURCE

LiteLLM Proxy · Envoy AI Gateway

PUBLIC SIGNAL

Uber GenAI Gateway

02 Инструментальный шлюз

TOOL / MCP GATEWAY

Слой между агентами и возможностями платформы.
Контролирует tool calls.

Agents

on-behalf-of user



Tool gateway

publish · authz · limits ·
audit · traces



Capabilities

API · runbooks

OPEN SOURCE

ContextForge · ToolHive

PUBLIC SIGNAL

AgentCore Gateway

03 Реестр возможностей

CAPABILITY REGISTRY

Источник истины: что можно вызвать и на каких условиях.
Описывает правила доступа.

Capability

owner · version ·
status



Access

scopes · limits · audit
links



Quality

eval sets · threat
cases

OPEN SOURCE

Backstage Catalog · DataHub · MCP Registry

PUBLIC SIGNAL

LinkedIn / Spotify

Слои сочетаются, но не заменяют друг друга. Подробнее: tellmeabout.tech

От API к capability map

Capabilities строятся вокруг workflows, не API

Platform product

Capability map

Intent map

Deploy, observe, secure, govern, approve, remediate.

Machine contracts

Tool schema, errors, side effects, cost, policy.

Registry

Owner, version, scopes, SLA/SLO, evals.

Доверие: read -> recommend -> act

Автономия дается на capability

Trust model

Read → act

Автономия

- Read: identity, quotas, telemetry.
- Recommend: plan, diff, risk, approval.

Где нужен контроль

- State changes, money or risk.
- Domain semantics: SLO, IAM, data.

Уровни зрелости agent-first IDP

Agent-first IDP

Maturity

От GUI-only портала к управляемой агентной поверхности с evals, telemetry и governance

L0	GUI-only ПОРТАЛ — ЕДИНСТВЕННЫЙ ВХОД	Сценарии живут только в GUI. API неполный, внутренний или нестабильный.	Нет безопасного вызова
L1	API / CLI ПРОГРАММНЫЙ ДОСТУП	Capabilities доступны программно, но контракты все еще написаны для людей-интеграторов.	API есть, agent UX нет
L2	Read для агентов УПРАВЛЯЕМОЕ ЧТЕНИЕ	Read-сценарии открыты агентам. Identity, квоты и telemetry включены с первого дня.	Read-only поверхность
L3	Recommend / Act ДОВЕРИЕ ПО РЕЖИМАМ	Рекомендации и действия идут через доменных агентов: policy, approval, откат, kill switch.	Только через gates
L4	Governance at scale МАСШТАБНЫЙ КОНТРОЛЬ	Evals, агентная telemetry и governance работают на масштабе. GUI становится control plane.	Автономия под контролем
Вектор зрелости: GUI -> API -> read -> recommend/act -> управляемая автономия.			

Модель угроз (threat model) для агентной разработки

Безопасность

Threat model

Prompt injection

OWASP LLM01: инструкция в тикете, логе или комментарии

АТАКА тикет просит игнорировать правила и вызвать tool

ЗАЩИТА текст = данные; actions через trust gates

Tool poisoning

MCP tool прячет инструкцию или меняет поведение

АТАКА описание tool подменяет поведение агента

ЗАЩИТА registry, owner, review; запрет обхода

Data exfiltration

read sensitive + write наружу

АТАКА данные уходят через web или ticket comment

ЗАЩИТА read/write split, PII redaction, egress allowlist

Overbroad retrieval

поиск приносит лишнее в контекст

АТАКА «искать по всем данным» обходит видимость

ЗАЩИТА права на каждом вызове от имени principal

Secrets in context

токены из логов попадают в следы работы

АТАКА секрет оседает в prompt, trace или history

ЗАЩИТА secret filters на gateway и leak evals

Confused deputy

агент действует с лишними правами

АТАКА просьба становится эскалацией привилегий

ЗАЩИТА on-behalf-of: права = пересечение

Часть 3

03

Что у нас уже есть

Building blocks: модели, инструменты, контекст, agent
mode и SDLC-агенты



**Saint
HighLoad++**
2026

Model gateway: доступ к LLM

Security и экономика

Платформа

Модельный шлюз

- One gateway без прямых provider-интеграций.
- Security: secrets, PII, data policy, audit.
- Quotas and cost accounting by team/scenario.
- Models onboard centrally; clients unchanged.

Tool gateway: MCP Hub/CLI

Платформа

MCP

Доступ к tools

- CLI, APIs, runbooks, service ops.
- MCP Hub: owner, scopes, schema.
- Controls: allowlist, on-behalf-of, audit, gates.
- Один интерфейс для IDE/agents.

Nessy: company-context copilot

Платформа

Nessy

Не просто Copilot

- RAG по codebase, docs, libraries.
- Знает conventions, ADR и golden paths.
- IDE и internal tools.
- Ответы знают текущее состояние систем.

Spirit: agent mode на платформе

Платформа

Spirit

Internal PaaS + agents

- VCS/CI/CD: shared policies, paths.
- Agent mode: ticket -> PR.
- Sage: AI-assisted troubleshooting.
- Agents share policy perimeter.

Наши общие агенты в SDLC цикле

Наш контур

SDLC agents

Это не один универсальный помощник — это набор доменных агентов на общей платформе, политиках и телеметрии

1 Разработка / IDE

- IDE agent mode
- Nessy Alpha
- Nessy CLI

2 Тестирование

- T-Cover: unit tests (~10% MR)
- T-Weaver: test cases из TestOps
- AI-выборщики для mobile MR

3 Ревью / качество

- AI Code Review
- API Governance: review OpenAPI

8 Инфраструктура

- Sandboxing агентов
- Запуск агентов в облаке, не на рабочей машине

ЕДИНЫЙ СЛОЙ

Agent platform

Шлюз моделей, политики, песочницы, телеметрия и стоимость.

4 Дизайн

- AI-линтер макетов
- figma2code

7 Миграции / данные

- SWE-агенты: миграции, автофикс билдов
- Data/DS-агенты, DeepResearch, AutoML

6 Эксплуатация

- SRE-Agent: RCA, аномалии, логи
- ИТ-поддержка: тысячи тикетов/мес

5 Безопасность

- Security-агенты
- AI SAST

04

Как мы это измеряем

Вместо vanity metrics смотрим на скорость, качество,
экономику, telemetry и реальное распределение
времени инженера



**Saint
HighLoad++**
2026

AI-код — плохая метрика

Управляем цепочкой от внедрение до economics

Анти-метрика

AI-LOC

Соблазн

- Легко собрать и красиво показать в отчете.
- Растет сама по себе вместе с внедрение.

Вред

- Мотивирует генерировать строки, а не ценность.
- Ничего не говорит о качестве, рисках и стоимости.

AI эффект на SDLC

Поставка, не LOC

Метрики

DORA × SPACE × DevEx

Process map

- DORA: lead time, frequency, recovery, CFR.
- SPACE/DevEx: flow, feedback, load, satisfaction.
- Adoption is input, not proof.

Signals

- onboarding time и первый MR;
- pipeline, testing, review time;
- rework, MR size, incidents, defects.

Evals/telemetry ВМЕСТО MAU

Metrics

Telemetry

Automation снижает GUI

- Trace: intent -> tools -> approvals.
- Span: user, agent, tool, model, cost.
- KPI: intent success, safe-change time.
- Evals regress agent-facing interfaces.

На что уходит время разработчика

Ускоряем весь день инженера

- Coding: 30–35%.
- Встречи и синки: 25–30%.
- Wiki, docs, context search: 12%.
- Ревью кода и обсуждения: 10–15%.

Reality check

Time split

05

Влияние на инженеров

Платформа сама по себе ничего не меняет — эффект дает перестройка процессов и командных привычек



**Saint
HighLoad++**
2026

Одна платформа — разный эффект

AI внедрение — социотехническая задача

Социотехника

Ключевое наблюдение

- Одинаковые tools дают разный результат.
- Эффект там, где меняется работа.
- Требования и взаимодействие перестроены.
- Платформа необходима, но недостаточна.

Что отличало успешные команды

Отличие — в перестройке работы

Кейсы

Сравнение

Без эффекта

- Включили ассистента, ждали ускорения.
- Задачи ставятся «по чату».
- Review/tests без изменений.

С эффектом

- Requirements стали agent contracts.
- Review адаптирован к AI-изменениям.
- Context/checks в pipeline.

Сопротивление — часть внедрения

Automation меняет рабочие процессы

- Fear of replacement реален.
- Работают обучение, прозрачные цели, quick wins.
- Team champions убедительнее приказов сверху.
- Пилот -> обратная связь -> масштабирование.

Люди

Сопротивление

День инженера: до и после

Меньше строк, больше intent/context/checks

Практика

До / после

До

- Пишет код руками.
- Контекст из чатов/wiki/calls.
- Review чужих diffs.

После

- Формулирует intent/context/criteria.
- Орkestрирует agents и parallel tasks.
- Валидирует результат, больше design.

Три вида агентов в дне инженера

Практика

Agent modes

Редактор, терминал и фон закрывают разные режимы работы — часто используются вместе

DEVELOPER DAY

Один день — три режима

Чем дальше агент от текущего курсора, тем важнее постановка задачи, границы доступа и проверка результата.

1 В редакторе

IN-FLOW / IDE

КОГДА ПОДХОДИТ

Когда инженер уже пишет код и хочет completion, быстрый edit или объяснение без выхода из IDE.

ПРИМЕРЫ

GitHub Copilot · Cursor · Windsurf · JetBrains AI Assistant

2 В терминале

GOAL -> TOOLS -> TESTS

КОГДА ПОДХОДИТ

Для multi-file задач, исследования незнакомого кода и работы, где агент запускает код и реагирует на результаты.

ПРИМЕРЫ

Antigravity CLI · Claude Code · Codex CLI · Open Code · Cline

3 В фоне

HANDOFF -> SANDBOX -> PR

КОГДА ПОДХОДИТ

Для понятных и хорошо ограниченных задач, которые можно отдать агенту на несколько часов: known bug fix, генерация test suite, миграция между версиями фреймворков.

ПРИМЕРЫ

Google Jules · GitHub Copilot agent mode · Cursor background agents · AlphaEvolve

Практика не сводится к выбору одного инструмента: один разработчик часто переключается между всеми тремя режимами в течение дня.

Роль инженера становится шире

Ценность — управление системой изменений

Роль

Leverage

Постановка

Цель, context, ограничения, DoD.

Оркестрация

Разложить, delegate, собрать результат.

Валидация

Проверить behavior, risks и intent.

Навыки-рычаги для инженера

Навыки

Skills

Чтобы агенты усиливали

- Context engineering: управлять контекстом агента.
- SDD: проверяемые contracts вместо размытых задач.
- Evals: измеримое качество agent work.
- Platform literacy: capabilities and boundaries.

Чего не происходит

Популярные страхи vs практика

Мифы

Reality check

- Инженеры не исчезают: judgment важнее.
- Джуны нужны; меняется обучение.
- Ревью кода проверяет contracts and proof.
- Ответственность остается на человеке.

Пять тезисов, которые стоит забрать

Главное из трех частей доклада

- Внедрение случился: управляйте новой физикой разработки.
- Эффект дает agent-first platform layer.
- Мерьте внедрение -> пропускную способность разработки -> quality/risk -> economics.
- Без изменения процессов инструменты не окупаются.

• [↗ polomodov.tech](https://polomodov.tech) • [↗ ai4sdlc-research.space](https://ai4sdlc-research.space) • [↗ tellmeabout.tech](https://tellmeabout.tech)

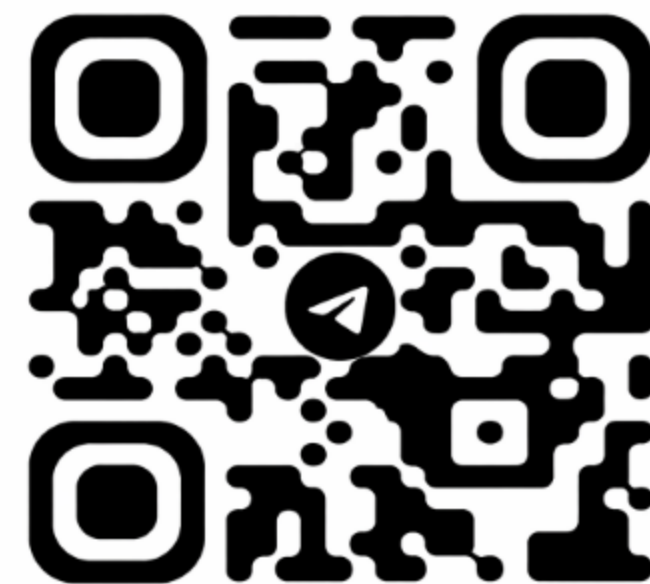
Книжный куб

Ссылки и материалы

Все ссылки и материалы к докладу собраны в Telegram-канале
"Книжный куб".

Александр Поломодов

Technical Director & Fellow,
Т-Технологии



@Book_Cube

Оставьте обратную связь



Александр Поломодов @ T-Bank



Saint
HighLoad++
2026