QAT GLOBAL
Quality • Agility • Technology



Agentic Software Engineering:

How to Scale AI Across the SDLC Without Losing Control

Turning AI investment into business results across the SDLC, without losing control

DIAMOND AI SOLUTIONS

STARTING POINT

The question behind every AI investment

Organizations are investing heavily in AI. Is it paying off?

Are AI investments delivering real business results?

Where the opportunity is

How applications are built and modernized.

AI is no longer just copilots

AI has moved beyond copilots, to specialized agents across SDLC.

What this session covers

Less time and cost, better quality, and higher ROI, without losing governance or human accountability.

And one more question: how do leaders guide the cultural change that makes it stick?

STARTING POINT

What is Agentic Software Engineering?

Specialized AI agents, coordinated by a structured method, doing defined work across the SDLC

Requirements → **Architecture** → **Development** → **Code review** → **Quality assurance**

AI copilot

Assists the developer

- Helps one developer, in one editor
- Suggests code or text
- The developer drives every step

AI agent

Executes the work

- Carries out multi-step tasks toward a goal
- Plans, acts, and uses tools
- Works within limits that people set

From assistance to delegation. IT CHANGES THE WAY YOU WORK!

STARTING POINT

Spec-Driven Development

The approved specification, not a chat prompt, is the contract that people and AI agents work from

THE SPECIFICATION CHAIN



↻ **Feedback from review and QA goes back into the specifications, so the next cycle starts better**

Lives in the repository

Simple text files next to the code, versioned and reviewed like code.

AI drafts, people approve

Agents build only from an approved spec. Guardrails keep them inside it.

A built-in audit trail

Every change traces back to a requirement, and to the person who approved it.

Without a spec, AI fills the gaps with assumptions. With a spec, it follows the contract.

WHY PILOTS STALL

The pilot that won’t scale

A realistic scenario

Stop Prompting

Start Delivering

CISO

Which credentials did the agents use?

CFO

What did it cost, and how will it grow?

Chief Risk Officer

Who approved the requirements? Can we prove the chain to an auditor?

CEO

If this works, why not everywhere?

The pilot is real. The evidence is not. This is an operating model gap, not a technology failure.

WHY PILOTS STALLS

Six questions leaders must answer

On demand, and with evidence

	Leadership question	Why it is hard today	What leaders should expect
CISO	**Which agent did this, under whose authority?**	Agents reuse developer sessions or shared accounts	**A unique non-human identity and a named human owner**
CISO	**What was the agent allowed to do?**	Broad permissions, inherited from whoever launched it	**Deny-by-default policies, checked before each action**
CRO	**Which requirement did this come from?**	Intent lives in chats, tickets, and memory	**A versioned chain from intent to code, tests, and review**
CRO	**Who reviewed and approved it?**	Informal approvals that vary by team	**Defined human gates from requirements to release**
CEO CFO	**What did it cost, and was it worth it?**	One invoice line, no link to value	**Cost per project, work item, and agent**
CRO	**Can we prove it to an auditor?**	Evidence scattered across tools and inboxes	**One central, retained audit trail**

If you cannot answer these on demand, you are not scaling AI. You are scaling exposure.

WHY PILOTS STALL

Individual prompting does not scale

Isolated prompt work speeds up tasks. It does not change the delivery system.

	Individual prompting		Agentic workflow
Who does the work	One developer, in one editor	→	**Specialized agents with defined roles and handoffs**
Source of truth	A chat prompt	→	**An approved, versioned specification**
Where intent lives	Chats, tickets, and individual memory	→	**A chain from intent to code, tests, and review**
Identity	Borrowed developer session	→	**Own agent identity and a named owner**
Rules	Policy by habit	→	**Written, enforced policies**
What improves	Individual tasks	→	**The whole delivery system**

Prompting is a skill, not an operating model. Scaling needs an agentic workflow.

WHY PILOTS STALL

Scaling is where most organizations stall

Share of organizations, by level of AI agent use



Single business function: no more than 10%

40%+

of AI projects predicted to be canceled by the end of 2027

Why:

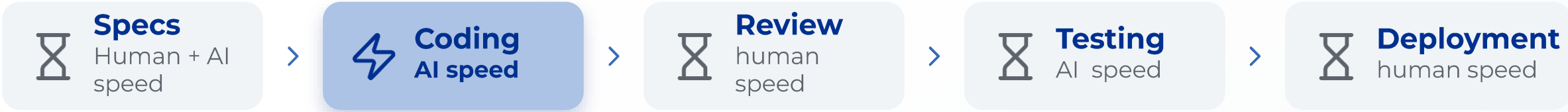
- Escalating costs
- Unclear business value
- Inadequate risk controls

Sources: McKinsey, The state of AI in 2025 (Nov 2025); Gartner press release (Jun 2025)

WHERE COPILOTS FALL SHORT

Spec-Driven Development is moving the bottleneck

Copilots speed up tasks, not the system around them.



The new way of working is focusing on the Specs!

Source: Anthropic, The AI-native SDLC playbook (Aug 2026)

Introducing QAT Global's Diamond AI



WHAT THE MARKET NEEDS

Eight core platform capabilities

One source of truth for AI-accelerated engineering, with complete control

VISIBILITY, COST, AND GOVERNANCE

1

Control & Observability

See what's happening

2

Intelligent Cost Management

Control AI spending

3

Governance & Audit

Accountability in every workflow

4

Context & Continuous Learning

Keep knowledge connected

SECURITY, DESIGN, AND ACCESS

5

Non-Human Identity & AI Policies

A governed identity for every agent

6

Security Monitoring & Data Protection

Protect what matters most

7

Fast UI Prototyping

Turn ideas into UI prototypes

8

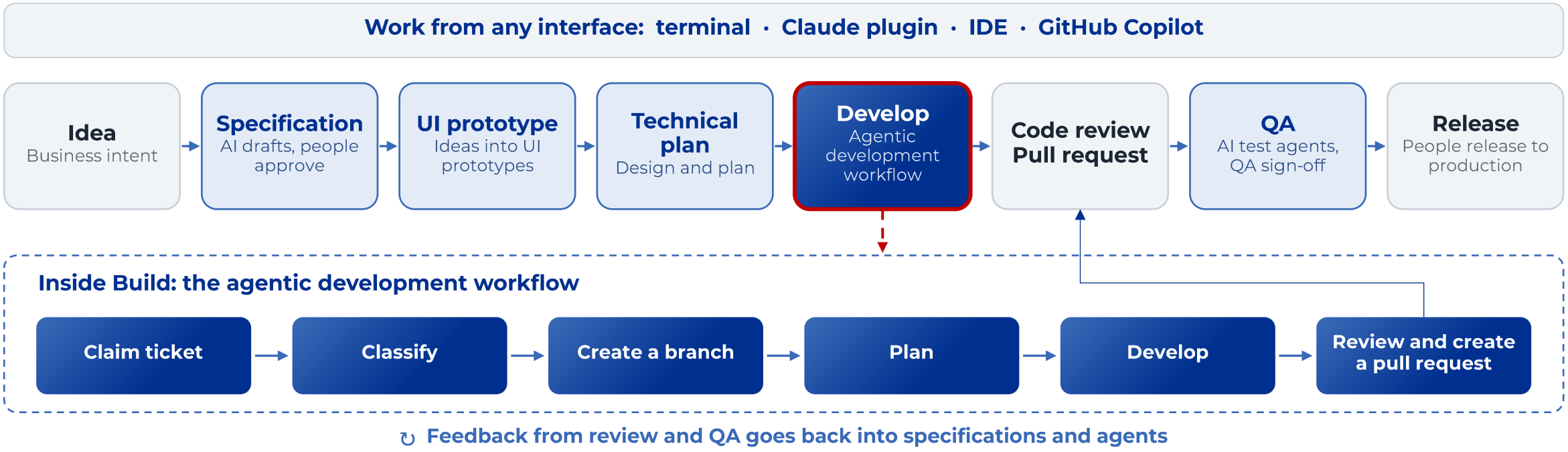
Multiple Interfaces

Terminal, Claude plugin, IDE, or GitHub Copilot

MAKING IT REAL

The Diamond AI flow, from idea to production

One governed path, with AI agents and people working together



One platform, one source of truth

- Control & observability
- Cost management
- Governance & audit
- Context & learning
- Agent identity & policies
- Security & data protection

AI agents do the work

AI agents + humans, in collaboration

Humans do the work

WHAT THE MARKET NEEDS

Our approach

Three decades of mission-critical applications. AI across the whole lifecycle, not only coding.

Quality before speed

Specialization over generalization

Multi-agent systems with defined roles

Validation at every step

Checked before work moves forward

Human-in-the-loop accountability

People own every key decision

AI Responsibilities

Prepares requirements and designs, and executes clearly defined engineering work.

People Responsibilities

Frame the problem, set priorities, approve the work, review the code, give QA sign-off, and control release.

One connected, governed path from idea to application. AI changes how fast work moves; people stay accountable.

A REAL PROJECT

The opportunity: how applications are built

AI across the whole lifecycle, not only the coding layer

Specifications
AI drafts, people approve

Code
AI generates, engineers review

Testing
AI agents test, QA decides

Bug fixing
AI speeds fixes, people verify

Build new capabilities faster

Reach a market window that would otherwise be missed.

Modernize existing applications

Move quickly, without giving up quality or governance.

Speed and accountability are not a trade-off. At every stage, a person directs the work and makes the decisions.

Source: QAT Global case study, qat.com/application-delivery-ai-speed (Aug 2026)

A real project: the challenge

A US financial services and payments company, January 2026

~5 months

to deliver work estimated at 18–24 months



A complete billing and payments system, built, tested, and running by June 30

Business goal

Add payments to a childcare billing product, and open it to a much larger market.

Hard deadline

Providers choose systems before the school year. Missing it meant waiting a full year.

Real complexity

Bills split between two households, government subsidies, and an audit-ready ledger.

Speed alone was not enough. On a billing system, being fast and being right must go together.

Source: QAT Global case study, qat.com/application-delivery-ai-speed (Aug 2026)

A REAL PROJECT

Business results: time, cost, and quality

Measured during the project, not estimated afterward

79%

shorter timeline: 5 months instead of 18–24

QA 2 Days

For a QA cycle that usually takes weeks

5 vs. 10

people, instead of a team of 10

3 days to hours

for one task, estimated vs. actual

Market result

Launched in time for the school season. The client then added four more phases.

Return leaders can see

Sprint reports compare estimated human effort with actual AI-assisted hours.

Faster, with less effort and a smaller team, and no compromise on quality or governance.

Source: QAT Global case study, qat.com/application-delivery-ai-speed (Aug 2026)

Leading the cultural change

What leaders must change to make it stick

1

Redefine roles

Engineers move to framing, design, review, and orchestration. AI does not replace them.

2

Treat specs like code

Version-controlled, reviewed, and approved, like any code change.

3

Share one workflow

Move from individual prompting to a team workflow everyone follows.

4

Measure the return

Compare estimated human effort with actual AI-assisted hours.

5

Decide faster

Keep a short decision cadence with stakeholders, so fast work does not wait.

6

Start with one project

Prove it on bounded work. Then expand with evidence.

Sources: QAT Global case study (Aug 2026); Gartner, software engineering trends (Jul 2025)

AI adoption is a journey, not a launch. The Key is INTENTIONAL!

An intentional adoption strategy turns AI potential into lasting value.

INTENTIONAL COMMUNICATION · RESEARCH, PLAN, IMPLEMENT, EVALUATE

R Research

Baseline sentiment, readiness and delivery pain points.

P Planning

Define audiences, messages, channels and success measures.

I Implementation

Deliver communications in rhythm with each transformation step.

E Evaluation

Measure adoption signals and adjust the approach continuously.

Steps to transform Organizational adoption

01
Organizational Readiness

Assess systems, data and teams

02
Leadership Alignment

Executive sponsorship and goals

03
Role-Based Enablement

Skills, roles and new ways of working

04
Pilot Teams

Prove the model on real work

05
Embed at Scale

Reinforce, sustain, widen adoption

06
Continuous Improvement

Measure progress, tune the model

Technology and organizational transformation move together.

MAKING IT REAL

Seven steps to operationalize

Run it as a governed program, not as a tool rollout

1. **Start with bounded, measurable work**
   One initiative, clear requirements and metrics
2. **Make the specification the source of truth**
   Approved, versioned specs before any code
3. **Give every agent an identity and owner**
   Retire shared credentials and personal tokens
4. **Enforce policy in code, not documents**
   Deny-by-default rules before every action
5. **Instrument before you scale**
   Central logging, dashboards, and alerts first
6. **Define human gates explicitly**
   Make them impossible to bypass
7. **Measure outcomes, not activity**
   Speed, stability, rework, and cost per feature

A governed program, with clear entry criteria

QAT GLOBAL
Quality • Agility • Technology

Thank you

Questions?

Talk to a QAT Global AI & Engineering expert.