

PRACTICE OF PROBING · POP2

Clarity in AI-Native Development

Produce assumption-free, well-bounded, predictable code, and evaluate AI output before it spreads.

For Developers | Eliminate rework, enhance capacity | Assisted by doSmartQA with Zoe

Why now

> **The bottleneck moves from writing code to shaping and probing it.**

What goes wrong

- Code looks correct but carries hidden dependencies and unstated assumptions.
- Edge cases go untested until they fail late.
- Late failures are costly to trace, and AI-generated code multiplies them.

What POP2 does

POP2 builds a probing discipline in your developers, so they model structure and behaviour before implementing, and evaluate AI output before it spreads through the codebase.

The shift

From

- Code that looks right but hides assumptions
- Edge cases discovered in system test or production
- AI output accepted without being probed
- Rework spent tracing late failures

→

To

- Code that is assumption-free and well-bounded
- Structure and behaviour modelled before implementation
- AI output evaluated before it spreads
- Quality built in, not patched on

How the engagement works

A consulting and advisory engagement in four strands, assisted throughout by the doSmartQA with Zoe platform.

1 · DIAGNOSTICS

See where you stand

Interviews and a survey with your developers, plus analysis of your specs, components, recent changes and defects found after handover.

A Diagnostic report, with suggestions and a plan of engagement.

2 · STRUCTURED LEARNING

Masterclass

Four 90-minute sessions with a HyBIST expert: probing at large, probing before you build, probing code and AI output, then developer testing and clean handover.

Applied to your own code and components.

3 · IMMERSIVE COACHING

Habits, daily

Two weeks of short daily nudges, each a habit to practise on real work with Zoe.

Probing becomes routine, not an event.

4 · EXPERT-LED IMPLEMENTATION

Make it stick

Weekly consulting sessions to coach, advise on process and track the agreed measures.

Embedded in your team's rituals.

Included: a one-year subscription to doSmartQA with Zoe, your Smart Assurance co-pilot (**Embrace** plan), to assist you on your probing journey, during the engagement and beyond.

© 2026, STAG Software Private Limited

With Zoe at every step

Zoe is your Smart Assurance co-pilot in doSmartQA: one companion, with a different role at each step, for a full year.

DiagnosticZone

Zoe as assessor

- Assesses how well the entities and modules you build on are understood
- Surfaces gaps and inconsistencies across modules

ProbingZone

Zoe as intelligent assistant

- Identifies what must exist before a feature can be built
- Identifies missing knowledge about the entities a feature touches
- Walks through situations to design developer-test scenarios
- Checks a component before it is handed over
- Analyses structural complexity and the impact of a change

PlayZone · PracticeZone · SkillZone

Zoe as coach

- Builds a probing mindset for AI-native development, and the skills behind it
- Turns shaping and probing code into daily habits

At a glance

FOR	LENGTH	MASTERCLASS	DAILY NUDGES	CONSULTING	PLATFORM
Developers	Minimum 6 weeks. Scoped with you.	4 × 90-minute expert sessions	About two weeks	Weekly, to the end of the engagement	**Embrace** plan One year included

What you receive

- A Diagnostic report with baseline and engagement plan
- An expert-led Masterclass: four 90-minute sessions
- Two weeks of daily habit nudges
- Weekly expert consulting and advisory
- A closing review against the baseline
- A one-year **Embrace** plan subscription to doSmartQA with Zoe, the Smart Assurance co-pilot

Part of the Practice of Probing suite

Each POP targets a critical point in the lifecycle. Together they make probing an enterprise capability.

POP1	POP2	POP3	POP4	POP5
Requirement Clarity Product Owner, Business Analyst	Clarity in AI-Native Development Developer	Smart Assurance QA practitioner	Insightful Management Project manager	Capacity Improvement Senior manager

Start with a Diagnostic

Find out what your code is quietly assuming, and what late failures are costing you.

Talk to us

© 2026, STAG Software Private Limited