

Modernize the software you rely on.

Senior engineering judgement. Modern tools. A clear path from inherited code to maintainable software.

Refactoring · Front-end modernization

End-to-end testing · CI/CD & DevOps

- A stronger foundation for change

Your existing software

Clearer
codebase

Critical
journey tests

Repeatable
releases

Illustrative approach · Scope follows the review

The business has moved on. The software needs to follow.

01 Changes keep getting harder.

A small feature touches too many parts of the system.
Estimates grow and releases slip.

03 The interface holds people back.

Important tools have become difficult to use, maintain or extend.

| A focused review helps distinguish what to retain, improve or replace.

02 Too much depends on one person.

The original developer has left, or the team needs a clearer view of how the system works.

04 Every release needs a workaround.

Manual checks and fragile deployment steps make routine updates difficult to trust.

Understand the system. Then improve it in stages.



Illustrative sequence. Deliverables, acceptance criteria and release arrangements are agreed for each engagement.

The judgement **around the code.**

AI helps us write and explore code faster.

Experience connects that work to the system, the people using it and the decisions your business needs to make.

One lead accountable for the technical work and for making the plan clear to your team.

01 **Code**

30+ years of engineering experience to review, challenge and improve what the tools produce.

02 **System**

Understand dependencies and the workflows a change could affect.

03 **Business**

Founder and CTO experience to connect technical priorities with the work that matters.

04 **Explanation**

A written recommendation and walkthrough your team can act on.

Legacy modernization in a Canadian telecom.

Recent work · Contentful system extensions

During his telecom consulting work, Andrew completed a full refactoring of legacy Contentful system extensions and redesigned their front-end UI.

Legacy refactoring

Front-end UI redesign

Enterprise environment

Andrew's consulting experience; presented without client endorsement or confidential implementation details.

The work

Modernizing an existing tool within a larger software environment.

The relevant experience

Understanding an inherited codebase, restructuring it and redesigning the interface people work with.

How it informs 1HAX

Start with the system and its users. Set a specific scope for the improvements.

Software Modernization **Review.**

A focused look at an agreed part of your software,
with a practical recommendation for what to do next.

Typically 1–2 calendar weeks

After access and scope are agreed.
Timing is confirmed before work begins.

Scope and fee agreed in advance.
You keep the findings and decide how to proceed.

What you receive

1HAX / Review

- ✓ **A risk map**
The technical risks within the agreed scope.
- ✓ **Critical business journeys**
The workflows to protect, and gaps in current checks.
- ✓ **Prioritized recommendations**
What to address first, with dependencies and trade-offs.
- ✓ **A priced first sprint**
A proposed scope, acceptance criteria and estimate.



1HAX

Toronto & remote

Senior experience. Direct responsibility.



Andrew Leschinsky

Lead engineer · 30+ years in software

Andrew leads the work, drawing on hands-on engineering and team leadership experience. Specialist contractors can join as the agreed scope requires.

Also built in-house

Custom Fairy Tales · Highlight Reel

Our own products, separate from client consulting.

We also take on scoped business process automation.

Request a consultation.

Tell us what your software does, what is getting in the way and what you need to change.

1hax.com

andrew@1hax.com

+1 647-885-4737

