

## We modernize legacy systems

We solve the most expensive software engineering challenge today: figuring systems out

Organizations are drowning in unchangeable systems, severely limiting their adaptability and value creation. The root cause is the approach to understanding systems: developers spend over 50% of their time manually reading through code, an unoptimized activity consuming half the development budget. Since systems are too large to read comprehensively, most information is inferred, leading to belief-based rather than fact-based decisions.

Testing had this problem, too

In the old days, testing used to have this problem, too. Traditional development handed code to testers who lacked system understanding and had to guess how things worked. Test-driven Development, introduced in 1999, initially seemed wasteful — writing tests before code. However, the resulting test suites became thousands of small, contextual tools that captured the system model and domain knowledge. This transformed testing from guesswork into an engineering discipline by embedding understanding directly into the tests. The lesson: apply testing's solution to development problems.

Our solution: thousands of contextual tools ... per system

Every question about a system boils down to retrieving information from it. To decrease the time to answer, we need tools. But as software systems are highly contextual, we need our tools to be contextual, too. These tools can provide a compressed perspective specific to every question thereby speeding up manyfold how we make decisions.

Moldable Development is a systematic way of programming through contextual tools that requires new skills and tools. In practice, we have seen many times a speed of improvement that makes modern AI approaches look pedestrian. That isn't to say that AI is not useful. It is, and combining AI with Moldable Development can lead to further improvements.

We built the technology for it, free and open-source

For Moldable Development to be practical, building an individual tool must be inexpensive and organizing and combining thousands of them must be seamless.

Glamorous Toolkit shows how that can work in practice and how programming through contextual tools introduces a new feedback loop: faster answers enable more questions which lead to better solutions.

We validated the approach extensively

We applied and tested Moldable Development for more than 15 years by solving hard problems in mission critical systems from various domains and built in diverse technologies. For example, for one Fortune 500 company, we took a critical legacy system that they thought would not be feasible to understand within 1 year and provided a working solution in 4 weeks.