beacon

Clearing the (Software) Traffic Jam

Now is the time to change how software is built.

Spring 2023

Beacon interviewed more than 100 software developers, managers, and executives across a range of industries about their software delivery challenges. With permission, we recorded, transcribed, and allowed an AI-powered Large Language Model (LLM) to summarize most of the interviews. We then married the feedback with our own experience and research on the rapidly-developing AI space to draw conclusions and make predictions that we're sharing here.

The Takeaways

1. Software delivery efforts are often **behind schedule, over budget, and of poor quality**.
2. **People issues** (management, stakeholders, trust) often eclipse technology issues.
3. Artificial Intelligence (AI) is **profoundly changing** the way software is developed.
4. Generative AI receives the majority of the attention, but **Interpretative AI offers even greater promise**.
5. In most cases, **less (of the right kind of) information is better than more**.
6. Mapping technology to business problems has always been **an imperfect process**.
7. AI can **translate, distill and interpret** information faster than humans.
8. But **humans remain critical** to the success of software projects.
9. People succeed when they have **the right information at the right time**.

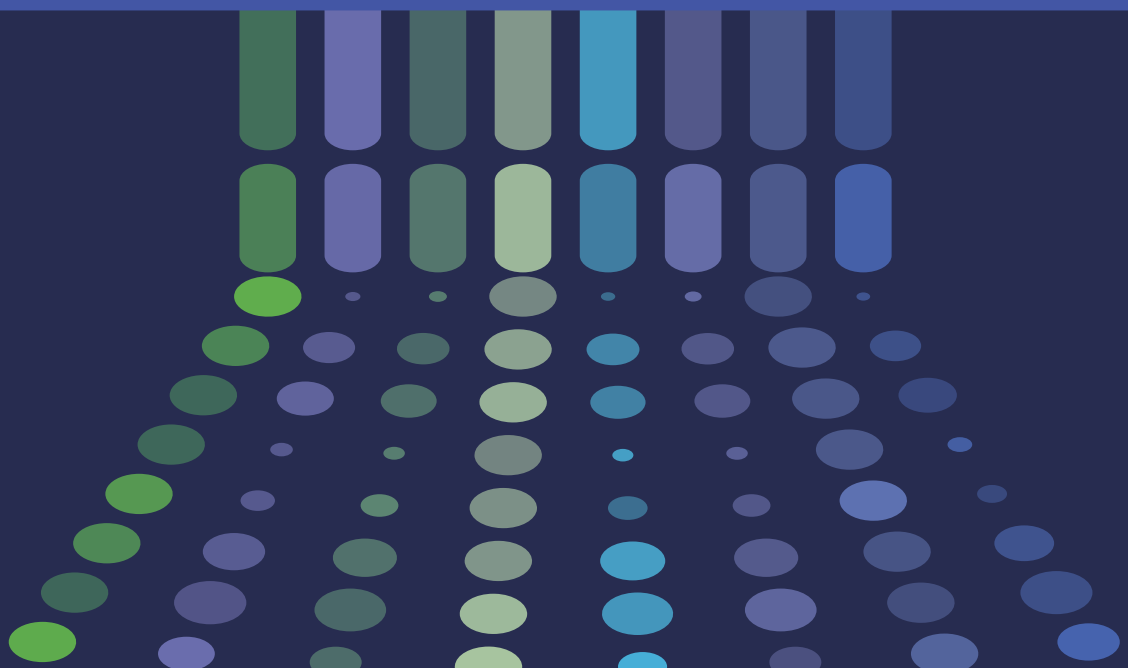


Table of Contents

Who We Are

Beacon is a software company, founded with the goal of improving the software development process. We are capitalizing on recent advances in Artificial Intelligence (AI), and marrying that with decades of software development experience to deliver a project intelligence solution that offers practical solutions to software and IT delivery challenges.

We are backed by Glasswing Ventures, a venture capital firm that funds companies harnessing the power of AI and frontier technologies to transform markets and industries.

You can find us at www.beacon.dev, and email us at hello@beacon.dev.

The Investor's Perspective

At Glasswing Ventures, we fund visionary entrepreneurs harnessing the power of AI and frontier technologies to transform markets and industries. Our world-class investing team is supported by industry veterans, as well as AI and academic advisors including the MIT Media Lab, Harvard Business School and the Carnegie Mellon Department of Computer Science, among others.

Technology is unpredictable and expensive to develop. By providing investment capital to exceptional teams, we give them the space to develop world-changing products. Along that path, they also obtain the customer feedback that is so critical to lasting success.

We continue to invest in Beacon because of their exceptional founding team, and their relentless focus on solving some of the hardest problems around the software development process. We've had more than a decade of experience with the founding team members across multiple companies, and can't wait to see how they will change the world!

RUDINA SESERI

Founder & Managing Partner
Glasswing Ventures

Executive Fellow
Harvard Business Schooll

Our Approach

We conducted more than 100 in-depth interviews with individuals responsible for software development projects, ranging from coders to managers and executives, to non-technical individuals responsible for project success. These people worked in many different sectors, including transportation, payments processing, healthcare, education, and technology, among others.

The structure of conversations broadly covered:

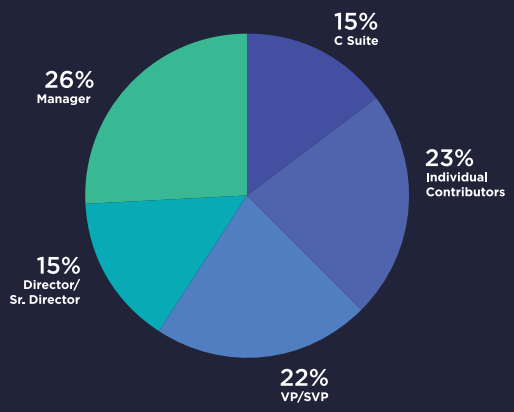
- The challenges of large, complex software projects
- The unique needs of front-line developers (coders)
- The role of leadership
- Ways that AI can and can't improve software development

Most interviews were recorded, transcribed, and run through an AI-powered Large Language Model for summary and distillation.

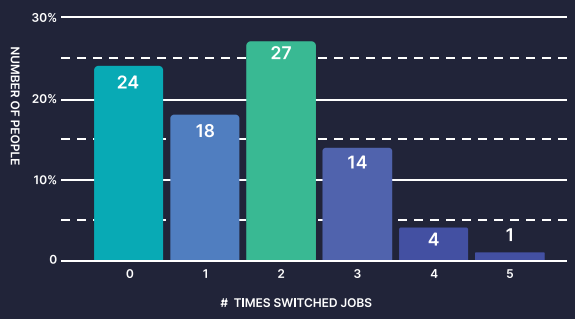
We express our gratitude to each of the people who contributed to this effort.

About Our Conversations

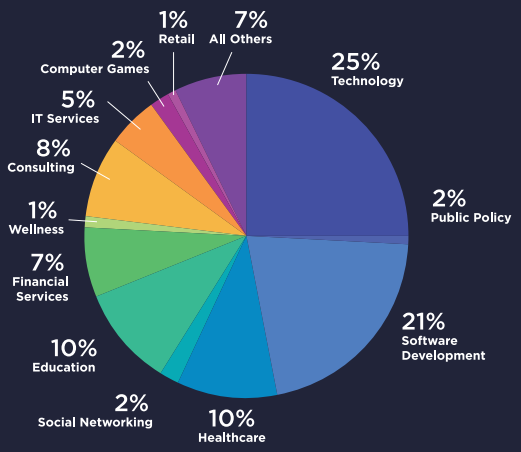
Roles



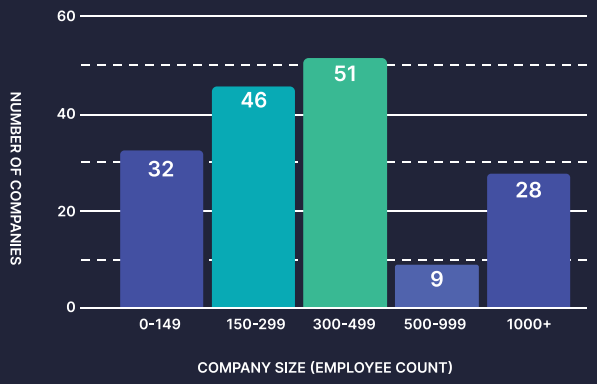
Number of Job Changes in Last 5 Years



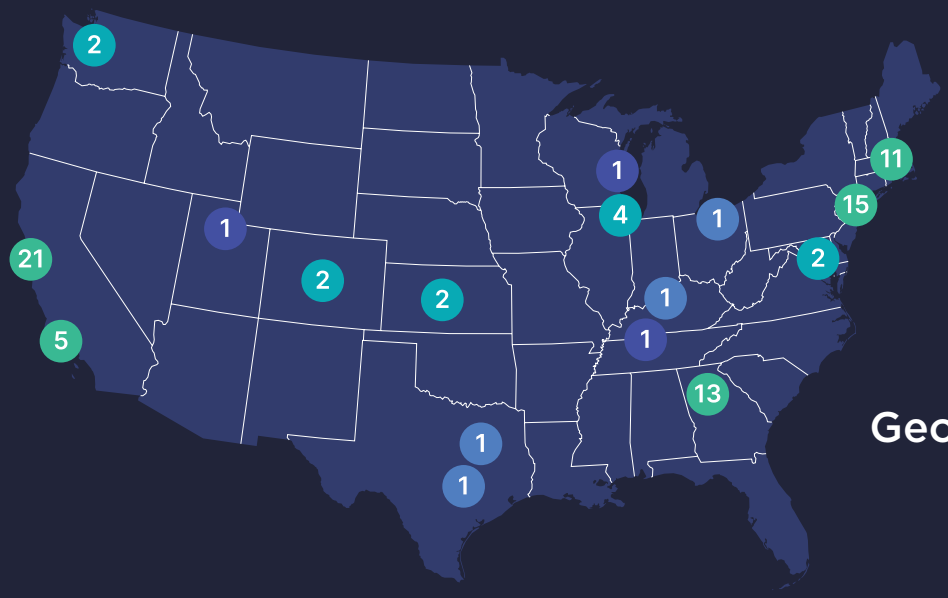
Industry



Company Size



Geography



What We Heard

People told us, often in vivid detail, of their software challenges, frustrations, and failures, both when developed in house and when working with outside consultants. The economic impact of those issues was obvious from the very first conversation, whether as simple as wasted hours and expanding budget or as complex as ruptured relationships that led to employee turnover and projects stretching multiple years beyond their original timeline.

The broad themes that emerged are unlikely to surprise those who have managed software projects. While each situation was unique, the problems were familiar and often well-documented.

In this section, we summarize four categories of major problems we heard, which will lead us to our perspective on potential solutions in the next section, "What we Predict."

1. People Issues Eclipse Technology Issues

Many of the people with whom we spoke were extremely technical. Even for this group, the majority of the air time in our conversations was devoted to non-technical issues, as this is where they wanted to focus. Their frustrations were around people, dynamics, and teams, and involved issues such as trust, mentorship, and conflict resolution. Particularly for outsourced development projects, there's an additional layer of stakeholders involved that amplifies this trend. Both direct managers, coders, and people managing these projects concluded how important people dynamics are to the ultimate output of the team, and therefore outcome of the project.

2. Process is Important

Regardless of whether software projects are completed by in-house teams or outsourced, the work is done by the coders themselves. Therefore, we spent a substantial amount of time with engineers on the "front lines" of software development to understand their day-to-day experience.

Front-line developers expressed:

- Frustration over poorly defined or ambiguous business requirements, both from internal teams and from external project teams.
- Frustrations with ineffective management.
- A sense that as information flowed down from management or from the client, it was diluted to the point of becoming unhelpful.
- A belief that a different process ("We should be more agile") would solve their problems.

Those feelings, applied across multiple stakeholders of a project, tend to inhibit open communication, collaboration, and transparency. The resulting defensiveness and transactional behaviors (e.g., "you need to tell me very clearly what you want, in detail, in writing") strongly correlated with poor outcomes.

Additionally, the focus on process, and most commonly Agile development, was of particular interest to the front-line developers. They tended to focus on this as an opportunity to improve outcomes, while executives and clients were more skeptical. As one executive told us, "agile feels to me like a way for project teams to avoid having to meet deadlines."

And even when Agile works as intended, it doesn't necessarily account for the downstream dependencies associated with meeting launch deadlines, keeping budget in check, training sales teams, building a marketing plan, or preparing customer support teams.

3. The Impact of Shortcuts is Insidious, and Cumulative

The phrase "technical debt" came up frequently, but is maddeningly ambiguous to define, and nearly impossible to measure. "Technical debt is code written by anyone other than me," was one cynical definition.

Cynicism aside, the cumulative impact of shortcuts, compromises, and decisions made under the pressure of looming deadlines was widely acknowledged, and described in terms such as "capacity drain" that over time costs money and slows feature development.

Particularly for development projects that are completed by outsourced teams, there's additional complexity to maintaining and updating systems.

This is well-supported in other research, especially in the public sector. One study found an estimated "70% of the government's IT budget is spent on maintaining legacy systems." And that "Just like an old car, maintaining these systems becomes more expensive as time goes by."[1]

4. Many Software Delivery Failures are Predictable

The perils of software development projects are no secret, and it's common knowledge that software delivery efforts are likely to be behind schedule, over budget, and/or of poor quality.

One outside study validated this, concluding that large ($6M+) government technology projects have a success rate of only 13%.[2] Another study concluded that only 29% of IT project implementations were considered successful, while 19% were categorized as complete failures.[3]

While many people are able to predict failure, few agreed that there were solutions at the ready. Amorphous solutions such as "better project management" and "more realistic deadlines" came up but are hardly actionable–especially given that the very mention of "more realistic deadlines" often translates to later project delivery, not earlier.

So the issues are long-standing, generally well-understood, but quite difficult to address. That leads to our predictions.

[1] "Doomed: Challenges and solutions to government IT projects" (The Brookings Institution)
[2] "Government tech projects fail by default. It doesn't have to be this way." (Harvard Kennedy School Belfer Center for Science and International Affairs)
[3] "Over 70% of tech projects fail but yours doesn't have to." (LeoCode)

What We Predict

As one prolific software developer said, "at this point, we're more limited by imagination than by technology." A bit of hyperbole, which we've heard a lot of recently, but it's likely founded. There is no shortage of excitement about AI, and it's poised to transform industries and society as a whole.

Microsoft co-founder Bill Gates calls AI one of two changes in technology that is truly revolutionary (the other being the graphical user interface, in 1980). Famous venture capital firm Andreessen Horowitz (a16z), somewhat jokingly, estimates the size of this AI market as "somewhere between all software and all human endeavors," but they're not wrong.

On a more pragmatic note, another venture capitalist rightfully notes, "AI will change everything we know, but it's hard to predict just how." Predicting the future is indeed hard, but we'll take our best shot.

1. AI is Already Improving Software Development

Much of the popular press is covering "Generative AI", a rapidly evolving subset of AI that can generate large amounts of new content based on the patterns and structures found in large datasets. ChatGPT, for instance, has gained significant recognition for its ability to understand and generate large quantities of human-like text in a matter of seconds.

More directly relevant to software development, GitHub's CoPilot, an AI-driven code completion tool, is revolutionizing the way programmers write and debug code. The use of tools such as CoPilot can free developers to focus on higher-level problem solving and more efficient work. CoPilot can complete up to 40% of a developer's code, and that number is poised to increase in the next five years. One can easily imagine the impact that will have on the efficiency and resourcing of software development projects.

> **"Software developers can understand business requirements, but business owners typically can't understand software code. AI can help bridge that gap."**

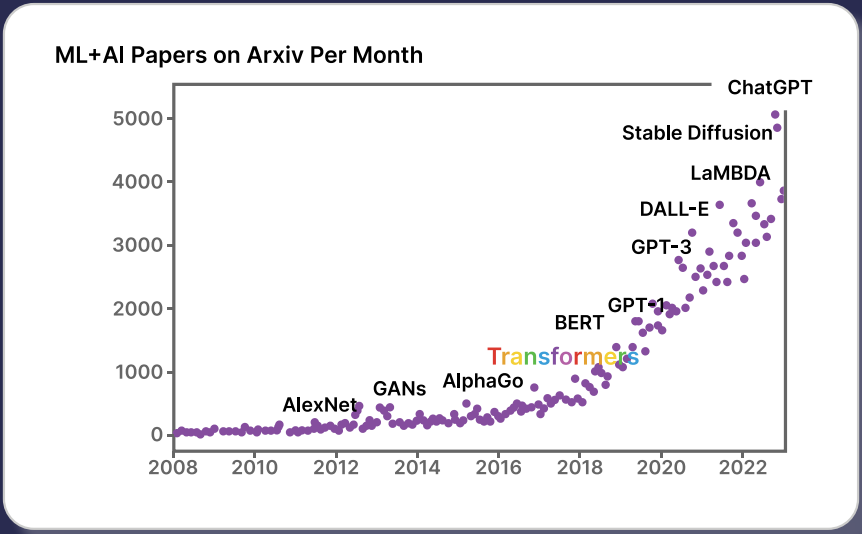
[4] "Cheating is All You Need" (Sourcegraph blog)
[5] "The Age of AI Has Begun" (GatesNotes: The Blog of Bill Gates)
[6] "Who owns the Generative AI platform?" (Andreessen Horowitz blog)
[7] "Every Company Needs an AI Strategy" (Sarah Guo blog)
[8] "Microsoft's GitHub Copilot AI is making rapid progress. Here's how its human leader thinks about it" (CNBC)

Why Now?

Research has been underway on AI for decades, and yet it feels as though we achieved a breakthrough only within the past few months. Judging by the number of papers written on Machine Learning (ML) and AI, the seemingly overnight nature of the breakthrough has been building for the past several years.[9]

AI, and the underlying Machine Learning (ML) research have benefited from recent advances on three separate dimensions: the sophistication of ML models; the increase in compute processing power, and the quantity of data available to the models.[10]

All three of these dimensions have been increasing rapidly, spurred by tens of billions of dollars in spending from tech titans including Microsoft, Alphabet (Google), and Meta (Facebook).



2. More is not always better

The obvious question with Generative AI is: “Who is going to read, or review, all of this newly-created content, and understand this code?” Enter the power of Interpretative AI (called “SynthAI” by some). The term emphasizes the importance of distilling (synthesizing) vast amounts of information, data, and code that we’re now able to create, at scale, and quickly.

In the world of software development, Interpretative AI offers the promise of bridging a long-standing gulf between “the business” and “the code.” Interpretative AI will enable companies to simplify and improve their workflows.

[9] “Every Company Needs an AI Strategy” (Sarah Guo blog)
[10] “The Generative AI Revolution Has Begun: How Did We Get Here? (ArsTechnica)

What We Predict (Continued)

3. Context Will Remain King

The next stage in the evolution of AI involves the incorporation of domain-specific data layers on top of existing AI models. The "how" of this very much remains an open question, but the goal is to provide depth and expertise akin to that of humans with extensive experience in a particular field.

To achieve this, AI has to move beyond large-scale, generic models and towards architectures that leverage multiple models. This new approach will include fine-tuned models trained on domain- and use-case-specific data sets, allowing AI systems to better understand and address the unique challenges and requirements of each domain.

By integrating these specialized models, AI technology can offer more targeted, context-aware insights, ultimately driving innovation and enhancing decision-making processes across various industries and applications.[11]

Well-crafted AI systems can tackle tasks that are difficult-to-impossible for humans, such as synthesizing and integrating **disparate data sources**. Effectively merging and synthesizing data from multiple sources, the AI delivers comprehensive insights for well-informed decision-making across organizations.

What About the Risks of AI?

Much that has been written about AI falls into camps that can broadly be described as either *utopian* or *dystopian*. We claim no special ability to predict the broader impacts of AI on society, but it does seem clear that the genie is out of the bottle.

AI promises enormous advances in productivity, and thus enormous profit potential for companies that can effectively harness it. For that reason, expecting individuals, companies, or entire countries to pause seems unrealistic.

As we incorporate AI into Beacon's product, we are doing so with the following potential risks in mind:

1. There's potential that applications of AI **move too fast** without understanding all of the potential consequences
2. Results generated from AI are not 100% accurate, which poses risk if we **fail to check its work**.
3. This risk increases if we don't include the appropriate **domain expertise** in AI applications.

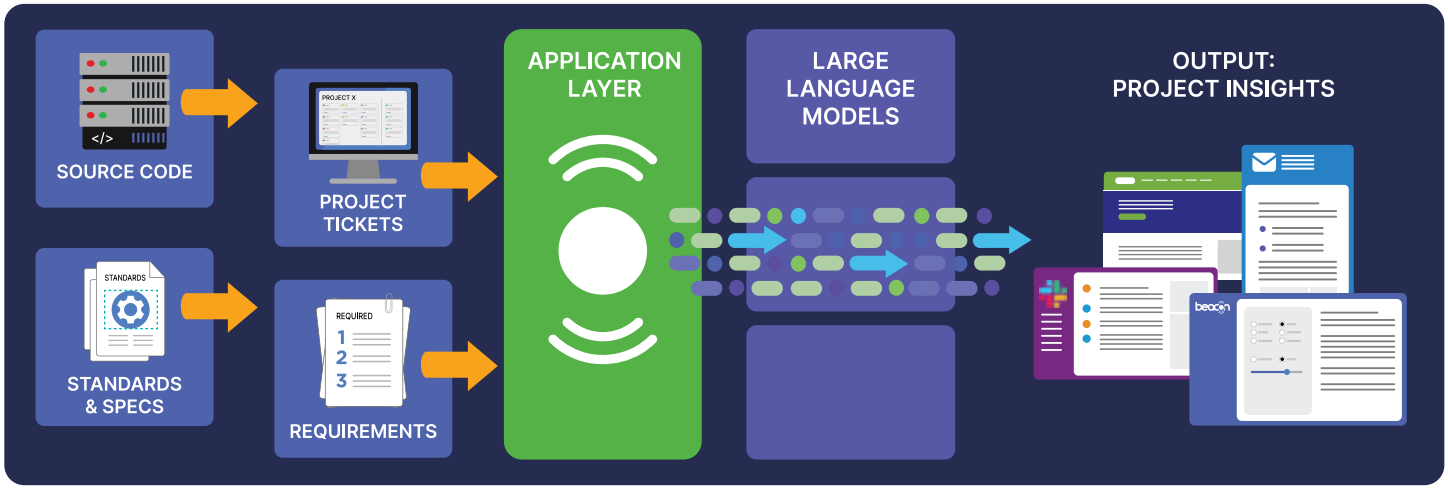
We are confident that regulation will catch up. Technology always outpaces regulation, but we can all innovate responsibly in the meantime.

[11] "For B2B Generative AI Apps, Is Less More?" (Andreesen Horowitz blog)

4. Success Will Continue to Depend on People

The typical Generative AI user interface creates long-form responses or code from concise prompts. Interpretive AI flips that, extracting concise output from massive amounts of data, ultimately transforming the way we leverage AI in decision-making processes.[12] The ultimate goal: to help humans make better, faster decisions.

For Interpretive AI to be useful, it must be context-aware and cognizant of its target audience. Understanding whether the synthesized information should cater to a high-level executive audience or provide more detailed insights for a technical audience. The process of distillation should be adaptable to the needs of various stakeholders, recognizing that the definition of distillation varies among individuals.



Final Thoughts

1. **Just get started.** Doing something with AI is far more likely to lead to success than simply thinking about it.
2. **Take small bites.** Focus = power. People are overwhelmed with information. Find ways to translate and distill information to help people make better decisions. There are significant advantages to be gained by solving problems in your industry.
3. **Be realistic.** It takes time to succeed, and today's reality may lag the hype. Applying consistent effort over time is a proven recipe for success.

[12] "For B2B Generative AI Apps, Is Less More?" (Andreesen Horowitz blog)

Got Links?

Use your phone's camera to scan the QR Code, taking you to an electronic version of this White Paper with clickable links.
Or visit beacon.dev/white-papers

beacon

Learn more at www.beacon.dev