Citation for the original published paper (version of record):

N.B. When citing this work, cite the original published paper.
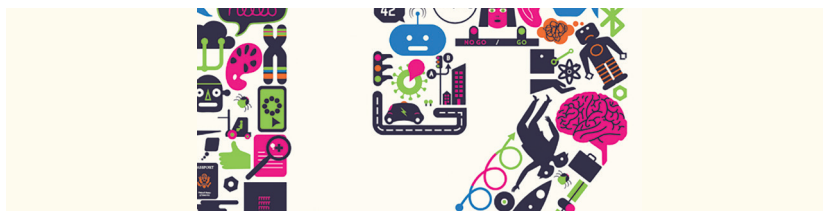
(article starts on next page)

Large Language Model Company Policies and Policy Implications in Software Organizations

Ranim Khojah , Chalmers University of Technology and University of Gothenburg

Mazen Mohamad , Research Institutes of Sweden and Chalmers University of Technology

Linda Erlenhov , **Francisco Gomes de Oliveira Neto** , and **Philipp Leitner** , Chalmers University of Technology and University of Gothenburg

// *The risks associated with adopting large language model chatbots in software organizations highlight the need for clear policies. We examine how 11 companies create these policies and the factors that influence them, aiming to help managers safely integrate chatbots into development workflows.* //

AI HAS REVOLUTIONIZED the toolbox of software engineers, allowing them to automate software creation, receive insightful recommendations, and perform a wide range of tasks.[1] In software organizations, the software product is gradually evolving to AI-powered software with the use of AI, more specifically, large language models (LLMs) in the development process.[2] LLMs are increasingly seen as valuable tools for improving productivity, which motivates enterprises to adopt them.[3]

However, these models have introduced risks and concerns that impact the organization, the software engineers, and the product. Integrating LLMs into software development raises challenges related to the quality and ownership of generated content,[4] which complicates accountability and can affect product reliability. In addition, interactions with LLMs (e.g., through external application programming interfaces) may expose organizations to liability if developers unintentionally transmit sensitive data, resulting in legal repercussions.[5] This risk can be amplified when developers use chat-based interfaces, which may create a false sense of humanlike familiarity that obscures security and privacy concerns, ultimately impacting trust.[6]

These concerns lead companies to either reject LLM adoption in development or implement policies to constrain LLM use. While policies are an effective way to mitigate these risks, the rapid evolution of these technologies means that defining clear rules and boundaries to guide their use in software engineering is urgent and unavoidable and also highly challenging. To better understand what software engineers,

managers, and decision makers need to consider when adopting LLM chatbots, we interviewed practitioners in management roles from 11 software organizations across four countries in Europe and Asia about their LLM policy (see "Data Collection and Methodology").

Why LLM Policies Are Needed

In many organizations, the push toward LLMs has been bottom up rather than top down,[8] in the form of a grassroots movement of developers experimenting with LLMs and chatbots to automate repetitive tasks and build software more efficiently. However, there are risks to such unstructured adoption of LLMs by individual engineers, namely, the lack of a clear plan and rules of engagement in the form of a clear policy.

During our interviews, managers expressed the concern that less experienced practitioners might not always critically assess the output generated by LLMs. This can lead to challenges in ensuring the quality and reliability of the product, and it may affect the level of trust and confidence between management and engineering teams.

> **Chief executive officer (EduCo):** "We noticed [that] some juniors took [the chatbot answer] too much for granted."

Moreover, managers almost unanimously raised the risk of software engineers unintentionally exposing intellectual property (IP) or sensitive customer data in their prompts to LLMs. Such incidents could violate data protection regulations like the General Data Protection Regulation, leading to legal penalties, loss of customer trust, and lasting damage to the company's reputation and financial stability.

> **Founder (ReaderCo):** "We have been afraid from the start that proprietary algorithms and stuff that we implement [are] indexed and used by chatbots."

Therefore, in response to the various risks raised, the interviewed managers recognized the need to establish policies, and they shared some of their approaches and steps in formulating and enforcing such policies.

How Industrial LLM Policies Work

We found that policies are framed by 1) the needs of the company (e.g., to minimize risks of compromising sensitive data) and 2) the opportunities chatbots offer, such as boosting

🔍 DATA COLLECTION AND METHODOLOGY

We spoke with 11 managers across 11 different organizations that use or have access to LLM chatbots (Table 1). The selection of managers was based on their authority to make or influence decisions regarding the use of LLM-based chatbots in software development at their organization. These chatbots fall into three categories: 1) self-hosted open source models and commercial closed source models with either 2) free subscriptions or 3) enterprise licenses.

Our participants held a range of roles—including team leads, process managers, and chief technology officers—which helped us capture diverse perspectives on how LLM chatbot adoption affects software engineering processes, teams, products, and customer value. Most of our participants worked in Sweden, but we also interviewed participants in Switzerland, India, and The Netherlands.

The interviews were semistructured, each lasting between 30 and 60 min. Participants consented after being informed about the study's purpose, anonymization, interview recording, and right to opt out. We analyzed the transcripts using thematic analysis. In an initial coding round, three researchers independently coded three interviews, with each researcher overlapping with another on two interviews to measure consistency. A total of 73% of the excerpts (166/225) overlapped among all the researchers, indicating agreement in coding, with most discrepancies involving excerpts where chatbots were used outside of development contexts.

Following this, the authors engaged in three collaborative sorting sessions. We reviewed and refined the codes, ultimately grouping them into broader themes. After three rounds, we reached thematic saturation, identifying key issues, such as organizational change and the creation of policies to guide LLM chatbot use. The interview protocol, list of extracted codes, and themes are available in our reproduction package to support future researchers, available at https://doi.org/10.5281/zenodo.15173862.
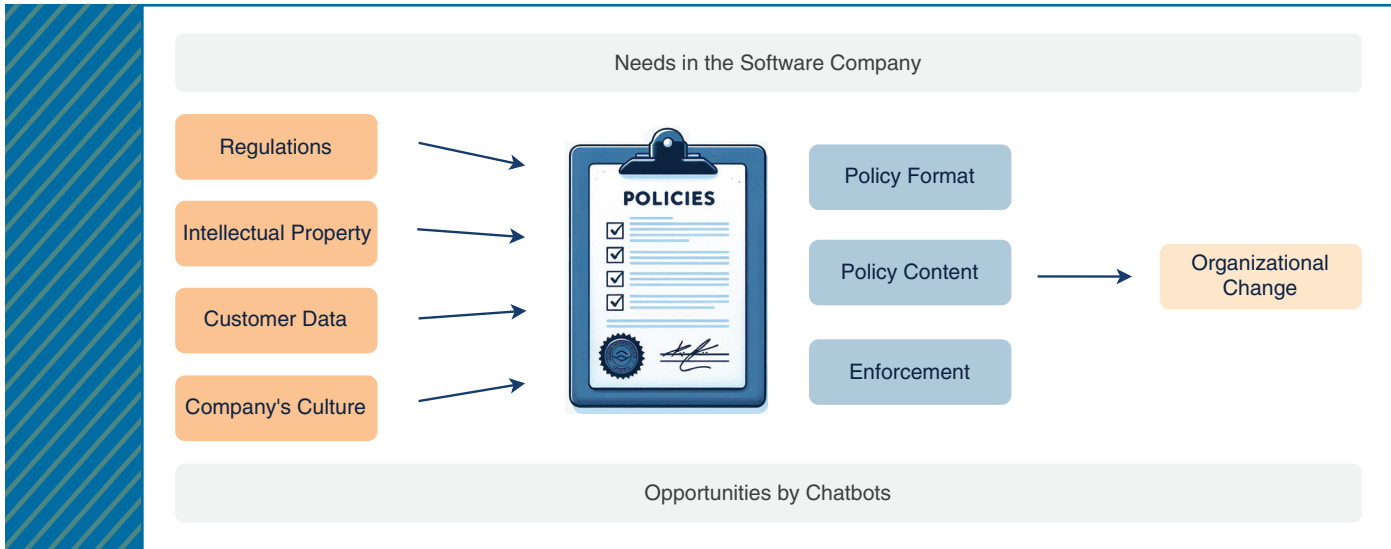
FIGURE 1. Factors that impact AI policy creation in software organizations. Note that the figure shows how the policies contribute to organizational change. While other factors may also lead to change, we focus on the specific role of policy within that broader context. (Source: Microsoft Copilot; used with permission.)

development team productivity. Our interviews revealed varied factors influencing policy creation, communication, and enforcement. While not necessarily exhaustive, these factors (Figure 1) reflect insights on the main components our participants identified as critical to shaping chatbot policies.

At the same time, policies are not just static documents. They are expected to influence ways of working, triggering organizational changes that, in turn, open up new needs and opportunities for chatbot use in software engineering, such as new roles within the organization. We discuss such organizational changes in the "Preparing Companies for an LLM Era" section.

Policy Drivers

Compliance with regulations and industry standards is a key driver of policy creation. Several participants specifically mentioned the European

Union (EU) AI Act,[a] which requires developers to clearly label LLM-generated and manipulated content. In the absence of chatbot-specific standards, three companies relied on ISO 27001[b] for information security that, among other requirements, demands classifying the data into sensitivity levels.

> **Manager (SysManCo):** "We did ISO 27001, and the big part of that work was to label the data … for what is sensitive and not sensitive (public, internal, confidential, critical data)."

We quickly found that the term "data" was too broad, and we needed to identify more specific levels of meaning to capture the nuances in how organizations assess what information can or cannot be shared with chatbots. For instance,

[a]https://artificialintelligenceact.eu
[b]https://www.iso.org/standard/27001

some policies focused on protecting IP by prohibiting developers from including code or requirements specifications in chatbot prompts.

> **Team lead (AeroCo):** "Our value is in our source code, so we try to keep that a secret."

However, for some companies, IP is not the source code but, rather, custom data they have collected or acquired. In these cases, the policy is more open to allowing sharing the code with chatbots but more restrictive with regard to sharing data.

> **Chief technology officer (EyeCo):** "Cursor indexes quite a large part of your code base, because then it works better because it has a larger context, and that is OK for us."

Regardless of the company's size or domain, our interviewees were in

agreement that customer data need to be treated with utmost care and that they are always considered sensitive.

Finally, we have observed that the size and domain of a company significantly influence its company culture and, consequently, the strictness of its policies regarding LLM usage. For example, micro- and small companies that rely heavily on open source code may not have a formal policy in place and trust their employees to use any chatbot, while larger companies or those in heavily regulated industries implement much stricter guidelines, going as far as blocking AI tools through their firewall.

Policy Creation and Enforcement

Key steps when creating a policy include 1) defining rules and guidelines to include, 2) deciding on the format for communicating the policy, and 3) a plan on how to enforce it. The content of the policy reflects important facets tied to the specific context in which the organization operates. Therefore, in this section, we focus on policy format and enforcement, and we discuss their content later.

The policy format varies across companies. Three participants reported using a formal policy document, while another company integrated the policy directly into the chatbot, including it in the terms and conditions shown in the chatbot interface. In most cases, however, companies (only) communicated their policies verbally, by e-mail, or through company-wide announcements. In such cases, the policies were concise and often reduced to a simple guideline, such as "do not share confidential information." Participants in smaller companies noted that this approach worked well and had not led to misunderstandings. EduCo suggests that this is mostly due to smaller team sizes

and shorter communication channels that allow for direct information sharing. Another reason for this informal approach is that drafting formal policies requires upfront research and, ideally, a legal team—a resource not always available to smaller companies. Two of the large companies we interviewed mentioned that their legal team handled the policy creation or that they outsourced the research and policy drafting to consulting firms.

> **Manager (ReaderCo):** "We don't have enough lawyers to speak to … or a legal team to get through this."

One challenge recognized by two companies was that if the policy is documented and stored elsewhere (e.g., on an intranet or shared drive), developers are likely to overlook it. To address this, they integrated the policy directly into the chatbot interface for developers to read before being granted access to the chatbot.

To promote employee compliance and ensure that employees stay up to date with policy changes, participants discussed different mechanisms for enforcement. The most common approach was to hold training sessions in which software engineers were taught how to apply the policy in practice and what types of uses were prohibited as well as how to use chatbots effectively. Managers generally found these trainings more effective than simply referring employees to the policy document.

> **Manager (ITServeCo):** "[The policy] is being circulated within the team members, and every six months we reconduct the training … for all new employees and juniors."

Policy training needs to go beyond traditional security training. It must also focus on responsible use, helping employees recognize potential risks, such as bias or misinformation, and developing skills, such as prompt engineering. At ITServeCo, one successful strategy was to conduct these trainings every six months and tailor the content to specific roles. For example, policy training for software testers focused on cases involving acceptance criteria and user stories.

Depending on how strict a policy item is, companies take different measures to enforce compliance. For example, ITServeCo prohibits using chatbots other than the ones it provides. To ensure compliance, it implemented technical controls, such as firewalls, and blocked access to external chatbot services. Another strategy reported is to take a lighter approach by encouraging employees to experiment with different technologies, offering special subscriptions to those who need access to external chatbots.

Some companies faced a more challenging scenario when it came to enforcing policies. They opted not to actively promote the use of chatbots in software development activities. However, they acknowledged that, even with clear restrictions, they had limited control over which tools engineers might choose to use independently. To address this gap while maintaining flexibility, two companies (AutoCo and AeroCo) offered safer alternatives, such as a licensed closed source solution or locally hosted chatbots. Although these options were made available to employees, they were neither explicitly recommended nor heavily promoted, largely due to high costs.

In general, we observed a trade-off between technical enforcement mechanisms (e.g., network restrictions and feature disabling) and other organizational strategies (e.g., offering alternatives and using trust-based models). While hard controls like firewalls are more enforceable, they may hinder innovation or lead to workarounds. In contrast, softer strategies provide flexibility but depend heavily on culture and employee cooperation with management.

In particular, the culture of the company plays a significant role in shaping how policies are created, communicated, and enforced. A culture that values open-mindedness and continuous improvement enables the company to learn from experience and adjust its policies or enforcement strategies as needed. The chief executive officer of EduCo summarized this perspective well when reflecting on areas where the company's own policy can improve.

Chief executive officer (EduCo): "We see mistakes more as a learning opportunity for everyone; there was also something wrong in our training or policies."

The contrasting strategies we observed highlight how both the policy format and enforcement must be tailored to fit the organization's culture and context. Standardized enforcement measures risk clashing with the organization's values and practices, reducing their effectiveness.

What LLM Chatbot Policies Cover

Regarding policy content, we observed a consistent presence of key elements in the policies of our participating companies, such as restrictions on shared data. However, the companies differed in the types of chatbots they used and in whether these chatbots were permitted for development-related tasks, such as code generation. These differences influenced which policy elements were prioritized and focused on by our participants. We group these differences into four usage and policy contexts: 1) nondevelopment usage only, 2) unlicensed closed source model, 3) closed source model with enterprise license, and 4) self-hosted open source model (see Table 1). By highlighting their focus, managers can better align the company's goals (e.g., complying with regulatory requirements or encouraging experimentation with new technologies) and more easily decide which specific rules to include in the policy document. Note that these contexts are not mutually exclusive; a single policy might cater to multiple contexts, with varying levels of detail, and a company may use several chatbots covered under the same policy. We aim to empower decision makers by raising awareness of these contexts, especially when adjusting current development practices.

Context A: Nondevelopment Usage Only

Policy focus: Define permitted and prohibited use cases for AI chatbots, explicitly disallowing development-related tasks.

Software organizations within safety-critical domains (e.g., automotive and aviation) centered their policy on how chatbots should and should not be used. In such cases, companies prohibit chatbot use to generate code contributions or restrict it to nondevelopment tasks, such as writing e-mails. Although software development involves many activities, we observed that our participants were cautious about allowing chatbots to assist with coding tasks in particular.

Manager (AutoCo): "We don't generate code or use [chatbots] in our code base at all."

To ensure that these policies are followed, companies need access to the interactions between employees and chatbots. Therefore, they prefer either hosting their own chatbot on internal servers or using commercial chatbots that provide licensed access to employees' prompts. Although they do not actively monitor the prompts' compliance with the policy, they want to store these data in case they are needed.

Context B: Unlicensed Closed Source Model

Policy focus: Restrict the types of data that can be shared to mitigate privacy and security risks.

Another context that we observed was when companies allowed the use of commercial chatbots (e.g., ChatGPT) in development without purchasing a license. The free subscription (no license) of closed source chatbots often allows providers to reuse interaction data to retrain their models, making this restriction critical for managers. As a result, we observed that the corresponding LLM policies focus mainly on the type of

data that is not allowed to be shared with the chatbot. These policies emphasized the need to prohibit sharing confidential information, such as customer data or IP.

Those restrictions also implicitly define how chatbots can be used. For example, if sharing production code is prohibited, activities like code repair or refactoring may not be feasible, whereas code generation could still be allowed. While stricter in companies without a license, such restrictions appeared in all policies (though in varying levels of detail), which reflects caution around legal, financial, and reputational risks, including potential impact on customer trust.

Context C: Closed Source Model With Enterprise License

Policy focus: Specify approved chatbots, and provide structured guidelines for access and setup.

In companies using enterprise-licensed chatbots, policies often focus on restricting employees to using only the chatbots officially provided by the company. Specific requirements for chatbot configurations are also commonly provided, e.g., disabling the option to share data with chatbot providers for further training or requiring employees to authenticate and access the chatbot through an internal portal.

Table 1. Information about our interviewees, their companies, the chatbots the companies use, whether the chatbots are used in software development activities, and the corresponding contexts.

Role	Company	Domain	Size	Country	Chatbot type	Use in development	Context
Chief executive officer	EduCo	Education	Micro	The Netherlands	Closed source (license)	Yes	C
Founder	ReaderCo	Reading technology	Micro	Sweden	Closed source (no license)	Yes	B
Team lead	CloudCo	Cloud platform	SME	Switzerland	Closed source (no license)	Yes	B
Chief technology officer	EyeCo	Eye tracking technology	SME	Sweden	Closed source (no license)	Yes	B
Manager	SysManCo	Systems management	SME	Sweden	Closed source (license)	Yes	C
Manager	TestCo	Testing consulting	SME	Sweden	Depends on customer	Yes	B, C, D
Team lead	ConsultCo	Software consulting	Large	Sweden	Closed source (license)	Yes	C
Team lead	AeroCo	Aviation	Large	Sweden	Open source (local)	No	A, D
Product owner	FlightCo	Aviation	Large	Sweden	Closed source (license)	No	A, C
Manager	AutoCo	Automotive	Large	Sweden	Closed source (license)	No	A, C
Manager	ITServeCo	IT services	Large	India	Open source (local) and closed source (license)	Yes	C, D

SME: small and medium enterprise.
For an explanation of the corresponding contexts, see the "What LLM Chatbot Policies Cover" section.
Company sizes were classified according to the categories recommended by the European Commission.[7]

> Policies allowing chatbot-assisted code generation will also reshape developer roles, potentially shifting the focus from writing code to verifying and composing it.

These policy measures allow the company to create a controlled environment that enables safe chatbot use, even without the full control it would have by hosting its own chatbot, which can be costly to implement and maintain for start-ups and small and medium enterprises.

Context D: Self-Hosted Open Source Model

Policy focus: Emphasize internal responsibility for verifying the correctness and quality of the model's output.

In cases where chatbots are hosted locally, companies can more easily enforce requirements (e.g., by not sharing a specific type of data) through system design rather than relying on users to comply manually. For instance, AeroCo built a custom chatbot with multiple environments, each tailored to different users' security clearances. Developers, for instance, access a restricted environment that prevents uploading or analyzing documents, while managers with higher privileges can use these features.

Another aspect that emerged, although not always formally stated in policies, was authority over chatbot usage. In some companies, junior engineers were initially restricted from using chatbots until they demonstrated responsible behavior and earned trust to access them independently.

When using chatbots for development tasks, engineers must ensure that the generated output is used safely to avoid harming the product under development. Most companies emphasized the need for code review and verification whenever an artifact (e.g., code, requirements, and test cases) is produced by an LLM chatbot. While all participants recognize the importance of verification, companies hosting models locally stressed it even more and noted that open source models typically underperform in code generation. This is also evident in modern code reasoning benchmarks, where the top results are often held by closed source LLMs, with the best open source models improving but still below closed source ones.[c] Currently, chatbot-generated results are verified using the same process as other artifacts created by an engineer or from a third party, such as testing or manual inspection.

Team lead (AeroCo): "If we introduce a third-party component or third-party code that we haven't written ourselves, it needs to go through quite rigorous testing before we can use it."

[c]https://www.swebench.com

Some companies pointed out that further research is needed to define additional verification steps tailored to chatbot output, which could then be incorporated into policies and help assure customers that the final product meets quality standards and delivers value.

LLM Policy Gaps

Two interesting gaps we noted were that none of the companies we interviewed addressed accountability (e.g., what happens if an employee violates the policy) or copyright concerns in their policies (e.g., how to ensure that chatbot output does not rely on copyrighted content). We speculate that this is partly because chatbot-related policies are still new and evolving. Interviewees explained that these topics were not included simply because such situations had not yet occurred.

Less attention to copyright concerns was also recently linked to loopholes in general AI regulations, such as the EU AI Act, which many companies rely on.[9] These loopholes appear to allow the use of AI-generated content even if it is trained on copyrighted data. For now, the legal situation remains unclear, and companies expect to update their policies as future court rulings and regulations emerge.

Preparing Companies for an LLM Era

The adoption of LLM chatbot policies is already driving organizational changes in software companies. Software process models, such as agile, include several activities like daily stand-ups or sprint retrospectives that strengthen team communication. These activities will become even more important with chatbot adoption, helping to maintain

team bonding and support informal knowledge sharing that chatbots cannot replace. Meanwhile, new activities related to chatbot governance emerge, such as monitoring prompts, promoting responsible use, and tracing LLM-generated artifacts. Those practices help enforce policies and compliance, which is feasible only when the chatbot is under company control, such as in local or enterprise-licensed chatbots , e.g., ChatGPT Enterprise.[d]

Currently, auditing chatbot usage often falls on managers, but ultimately, a new role for chatbot governance may be required. Policies allowing chatbot-assisted code generation will also reshape developer roles, potentially shifting the focus from writing code to verifying and composing it.[10] Unlike reviewing a single merge request, developers might need to evaluate multiple LLM-generated solutions across different prompts. Interestingly, despite public debate about chatbots replacing software engineers, our interviews suggest the opposite: chatbots create greater demand for engineers, though equipped with new skills.[6]

> **Manager (ITServeCo):** "We have adapted existing roles [of engineers], but we are adding members to our team to take care of the work that these [engineers] are doing. … AI has added members to the team since we have a lot of work going on because of the updates and other steps."

To support this shift, companies are starting to design specialized

[d]https://help.openai.com/en/articles/10875114-user-analytics-for-chatgpt-enterprise-and-edu-public-beta

ABOUT THE AUTHORS

RANIM KHOJAH is a Ph.D. candidate at Chalmers University of Technology and University of Gothenburg, 417 56 Gothenburg, Sweden. Her research interests include human–chatbot interactions in software engineering. Khojah received her licentiate of philosophy in computer science and engineering from Chalmers University of Technology. Contact her at khojah@chalmers.se or www.ranimkhojah.com.

MAZEN MOHAMAD is a researcher at the Research Institutes of Sweden and a lecturer at Chalmers University of Technology, 417 56 Gothenburg, Sweden. His research interests include security assurance, combined safety and security analysis, AI in software engineering, and AI for cybersecurity. Mohamad received his Ph.D. in software engineering from University of Gothenburg. He is a Member of IEEE. Contact him at mazen.mohamad@ri.se or www.mazenm.com.

LINDA ERLENHOV is a lecturer at Chalmers University of Technology and University of Gothenburg, 417 56 Gothenburg, Sweden. Her research interests include human aspects of software engineering and software development tooling. Erlenhov received her licentiate of engineering in computer science and engineering from Chalmers University of Technology. Contact her at linda.erlenhov@chalmers.se.

FRANCISCO GOMES DE OLIVEIRA NETO is an associate professor of software engineering at University of Gothenburg and Chalmers University of Technology, 417 56 Gothenburg, Sweden. His research interests include automated software testing and (AI) bots to aid software engineers. Oliveira Neto received his Ph.D. in computer science from the Universidade Federal de Campina Grande. Contact him at francisco.gomes@cse.gu.se.

PHILIPP LEITNER is an associate professor of software engineering at Chalmers University of Technology and University of Gothenburg, 417 56 Gothenburg, Sweden. His research interests include empirical software engineering, with a focus on software performance optimization and the development of web- and cloud-based systems. Leitner received his doctoral degree in business informatics from TU Vienna. He is a member of the Association for Computing Machinery. Contact him at philipp.leitner@chalmers.se or https://icet-lab.eu.

training aligned with their chatbot policies and practitioner roles (e.g., testers, developers, and managers). At ITServeCo, targeted training programs have already improved employees' perceived productivity and efficiency.

Through interviews with managers across a variety of companies, domains, and roles, we captured a broad view of the key factors and priority areas that shape LLM policies in industry. Managers and decision makers can draft policy documents that reflect their own company's context and culture, guided by the real-world practices of the companies we interviewed. Documenting such LLM policies helps companies navigate the era of AI-driven tools with greater clarity and avoid becoming overwhelmed when faced with important decisions regarding LLMs and software development.

References

1. R. Khojah, M. Mohamad, P. Leitner, and F. G. de Oliveira Neto, "Beyond code generation: An observational study of ChatGPT usage in software engineering practice," *Proc. ACM Softw. Eng.*, vol. 1, no. FSE, pp. 1819–1840, 2024.
2. A. E. Hassan et al., "Rethinking software engineering in the era of foundation models: A curated catalogue of challenges in the development of trustworthy FMware," in *Proc. 32nd ACM Int. Conf. Found. Softw. Eng. (FSE)*, New York, NY, USA: ACM, 2024, pp. 1819–1840.
3. S. Sharma, G. Singh, N. Islam, and A. Dhir, "Why do SMEs adopt artificial intelligence-based chatbots?" *IEEE Trans. Eng. Manage.*, vol. 71, pp. 1773–1786, 2024, doi: 10.1109/TEM.2022.3203469.
4. I. Ozkaya, "Application of large language models to software engineering tasks: Opportunities, risks, and implications," *IEEE Softw.*, vol. 40, no. 3, pp. 4–8, May/Jun. 2023, doi: 10.1109/MS.2023.3248401.
5. S. Herbold, B. Valerius, A. Mojica-Hanke, I. Lex, and J. Mittel, "Legal aspects for software developers interested in generative AI applications," *IEEE Softw.*, vol. 42, no. 2, pp. 68–75, Mar./Apr. 2025, doi: 10.1109/MS.2024.3476677.
6. N. Nahar, C. Kästner, J. Butler, C. Parnin, T. Zimmermann, and C. Bird, "Beyond the comfort zone: Emerging solutions to overcome challenges in integrating LLMs into software products," in *Proc. 47th Int. Conf. Softw. Eng.: Softw. Eng. Pract. (ICSE-SEIP)*, 2025, pp. 516–527.
7. "Internal market, industry, entrepreneurship and SMEs." European Commission. Accessed: Apr. 6, 2025. [Online]. Available: https://commission.europa.eu/about/departments-and-executive-agencies/internal-market-industry-entrepreneurship-and-smes_en
8. S. Lambiase, G. Catolino, F. Palomba, F. Ferrucci, and D. Russo, "Investigating the role of cultural values in adopting large language models for software engineering," *ACM Trans. Software Eng. Method., early access*, Mar. 21, 2025, doi: 10.1145/3725529.
9. J. Rankin, "EU accused of leaving 'devastating' copyright loophole in AI Act," *The Guardian*, Feb. 19, 2025. [Online]. Available: https://www.theguardian.com/technology/2025/feb/19/eu-accused-of-leaving-devastating-copyright-loophole-in-ai-act
10. M. R. Lyu, B. Ray, A. Roychoudhury, S. H. Tan, and P. Thongtanunam, "Automatic programming: Large language models and beyond," *ACM Trans. Software Eng. Method.*, vol. 34, no. 5, pp. 1–33, 2024, doi: 10.1145/3708519.