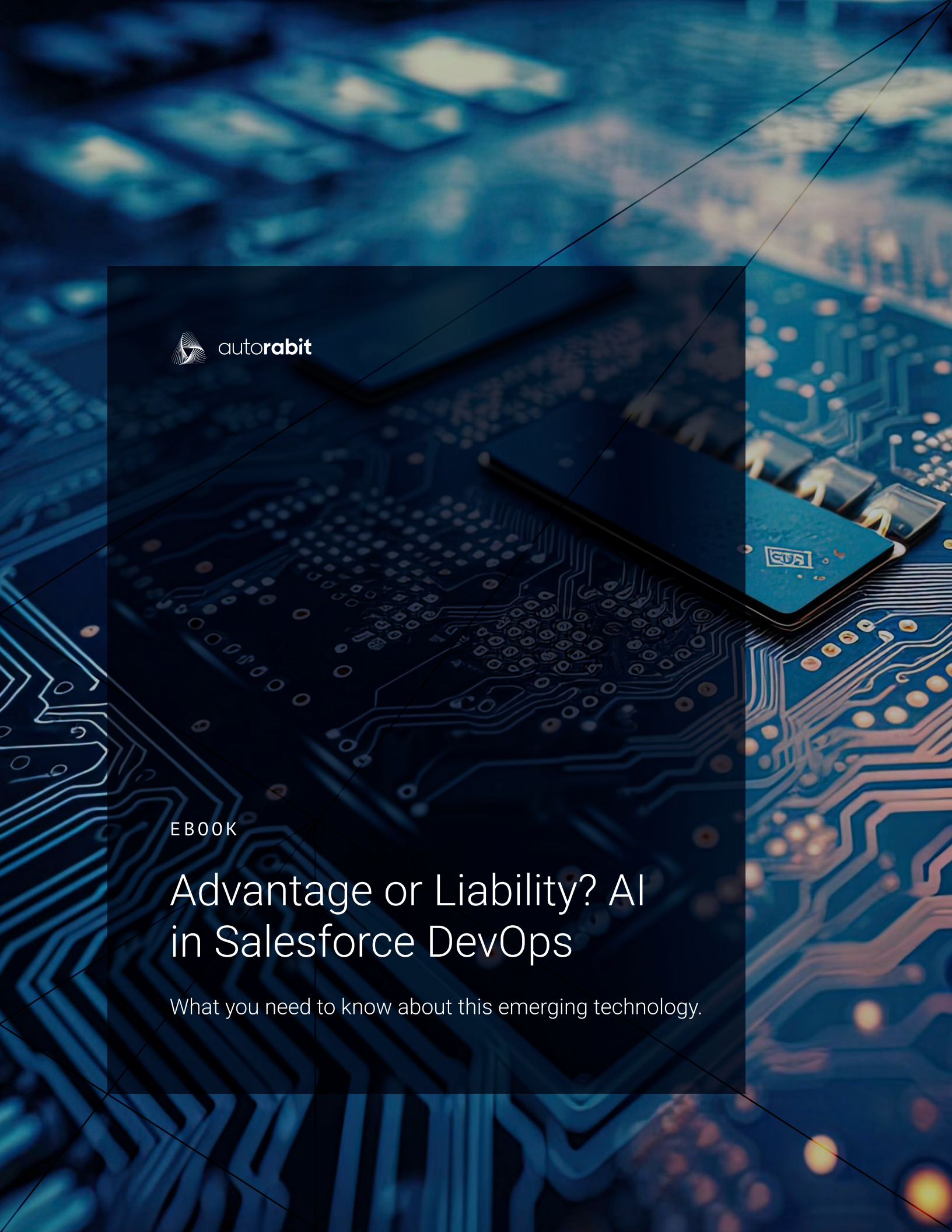


autorabit

EBOOK

Advantage or Liability? AI in Salesforce DevOps

What you need to know about this emerging technology.

INTRODUCTION

AI is everywhere you look these days. Recent advances in technological maturity are impacting everything from image generation and editing to writing high school essays. However, these aren't the only applications of this software.

Salesforce DevOps has grown to rely on strategic automation to streamline processes, reduce the strain on team members, and increase reliability. Recent expansions in machine learning capabilities have already impacted the way we write and review code changes.

Adopting new technologies can be an uncomfortable process for large organizations. However, failing to do so will create unnecessary challenges for your continued growth and expansion of offerings.

Earlier this year, Salesforce announced a new product: Einstein GPT. This product promised to "transform every customer experience with generative AI," according to the announcement.

But even though this product is still a pilot program, the implications are clear—things are about to change.

Now is your opportunity to take stock of what's on the near horizon and make critical preparations to avoid unintentionally creating problems. There are a lot of potential benefits to introducing this technology, but if it isn't done right, you also run the risk of introducing problems.

Let’s look at these 3 major aspects of AI in Salesforce DevOps:

01

Identifying Where We Are

LARGE LANGUAGE MODELS + GENERATIVE AI

Generative AI is constantly featured in the media lately. You most likely have some experience with it as well. ChatGPT is probably the most well-known version of generative AI, which is characterized by users providing a prompt the software uses to create content like images, videos, or text.

Potential growth using Large Language Models (LLM) is more applicable to Salesforce DevOps. It is very similar to generative AI in that it uses prompts, but it is more specialized in natural language processing. This means it can be used to generate lines of code.

New, open-sourced models are coming out all the time. However, the amount of input a model receives directly correlates to its reliability. Models that have been around for a while have more information to draw from and are better suited to writing reliable code and providing predictive analytics—capabilities that will continue to mature.

Here are some well-known LLM and generative AI tools available right now:

OPEN AI CHATGPT

GOOGLE BARD

MICROSOFT COPILOT

AMAZON CODEWHISPERER

BEDROCK AI

SALESFORCE DEVOPS TOOLS

Salesforce is in the process of introducing transformative new AI tools to address various aspects of the DevOps pipeline. And while these features aren't currently available on a large scale, they have been introduced in an AI road map.

Here's what we can expect to see from Salesforce in the near future:

Einstein Prediction Builder

Create custom predictive models that can predict customer behavior, optimize marketing campaigns, and identify leads without writing a single line of code.

Einstein Analytics

Gather insights and visualizations in real time to create customer dashboards and reports. These help organizations understand their customers better and make stronger data-driven decisions.

Einstein Language

Analyze and improve comprehension of customer inquiries with a natural language processor. This creates custom virtual assistants and chatbots that offer personalized support.

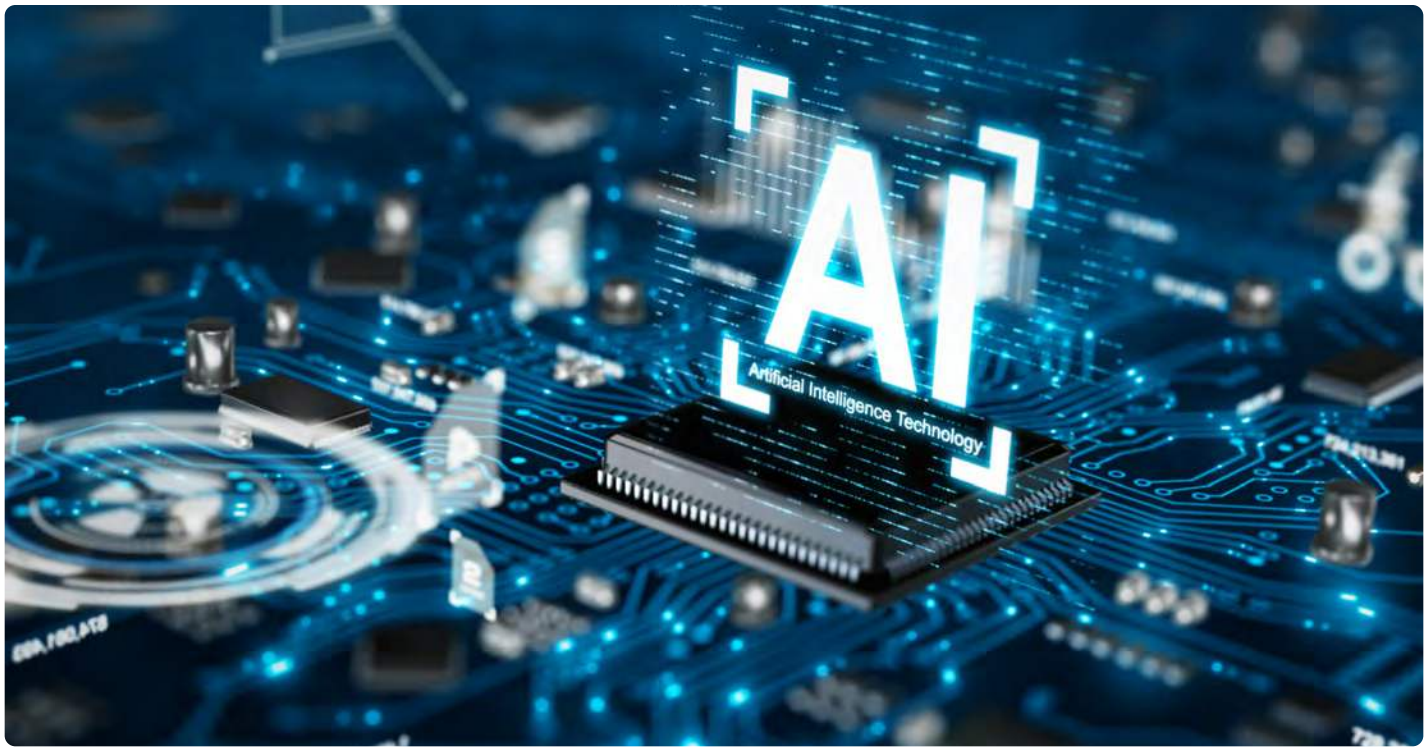
Einstein Vision

Evaluate and classify images through image recognition software. Create custom applications to perform recognition tasks and identify specific objects such as products and faces.

Einstein Discovery

Gather insights on business trends through the intelligent analysis of data. Improve customer relations and optimize business processes through the creation of custom predictive models and recommendation engines.

WHY EVERYONE'S EXCITED



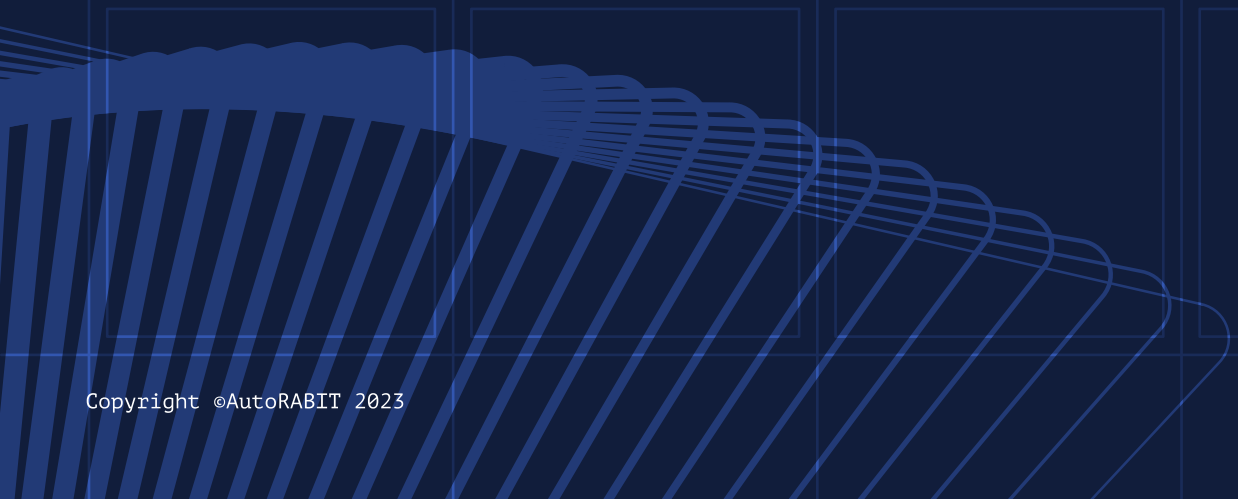
The benefits of utilizing LLM and generative AI tools are impressive. Consider how quickly your team members could produce new applications and updates if they could input a prompt and have a tool write the code? Efficiency would skyrocket as your team would only need to review this code to ensure proper functionality.

This technology has the potential to streamline development processes and introduce previously unseen levels of automation. This drastic reduction in tedious, manual tasks gives your team members time to go back through the generated lines of code and focus on creative manipulations, since the basic building has already been addressed.

The possible adaptations of this technology are limited only by the imagination. For instance, it could be used in conjunction with an organization's ticketing system. A user could submit a support ticket that would be routed to an AI model that could directly address their concerns. This is just one example of a possible use for this limitless technology, but it also highlights the capacity for changing how many business processes operate.

02

Understanding the Challenges



UNRELIABLE RESULTS

LLM and generative AI work base their responses and output on the information fed to them. This is known as a probabilistic model, which means they won't produce the exact same outcome every time. And while this is good to make these tools seem more conversational, it makes them unreliable in writing code.

This type of infrastructure sets up a feedback loop—information is fed into it that creates responses to given prompts, which are fed back into the repository from which it draws results.

Basing responses on open-sourced material degrades the reliability of the results. The model draws information from everyone who interacts with it—whether they're beginners or experts. Bad source data leads to bad content.

This can lead to what are referred to as *hallucinations* in the code. These are often the result of incorrect assumptions, logical errors, and flaws in the coding structure. They occur when the code is written in a way that assumes an action will occur even though it isn't present in the code itself. This leads to crashes, bugs, and other errors.

Spaghetti code is another potential result of these errors. This refers to code that isn't structured properly and becomes overly complicated and difficult to understand.

SECURITY AND COMPLIANCE CONCERNS

The biggest problems with LLM and generative AI in relation to regulatory compliance are the unknowns. Who owns the data fed into the system? Where is this data stored? Who can access this data?

Programs like ChatGPT use the input data to further train and refine the program. There have already been instances where this has led to data leaks.

Regulated industries need to maintain strict control over their data—including their code. Now imagine if a developer creates lines of code through an AI application—is that code stored in a public repository? If that code ends up in a company's production instance, a data security liability is created because that code is also living in an open-source environment.

These types of questions don't currently have an answer, but they must be considered before companies can safely introduce this technology.

LLM and generative AI tools are trained from public information—the code comes from GitHub. So if the code put into the tool teaches it that a particular pattern is the right way to solve a certain type of problem—and that approach turns out to have a security vulnerability—then every user who generated this coding approach has the same vulnerability.

We've discussed how LLMs can lead to accidental introductions of security risks, but intentional spiking of the source repositories is also possible. This is what's known as *prompt injection*. This is when a bad actor introduces problematic prompts into an LLM with the goal of creating malicious results that makes the models misbehave in various ways.

This technology has the potential to create unprecedented systemic risks.

IMPLEMENTATION DIFFICULTIES

Integrating new tools into your DevOps pipeline will always have a period of awkwardness. Team members aren't familiar with it. Processes haven't been perfected. This strange period might be short, but it always exists.

The problem with LLM and generative AI tools is that they essentially strap a rocket to your DevOps processes. Everything moves much faster and if your system isn't prepared, you run a higher risk of making some pretty damaging mistakes.

These tools are largely automated but only after a team member enters a specific prompt. And like anything else, these prompts will take some time to perfect. Initial entries might not be specific or refined enough to accomplish your goals—which can lead to bugs if they're not caught.

This is where quality gates come in. There will always be a need for safety nets that double-check the work of AI tools because unless you review the results, you never actually know what's in there.

These mistakes are likely to become less frequent over time, but they will never be totally eliminated. Even with highly automated tools, manual errors have the potential to create data security and quality issues.

Ensuring consistency across teams is a major factor in your success using these tools. Failing to communicate best practices and provide the necessary testing tools can result in bad or malicious code entering your repository.

03

Operating Safely

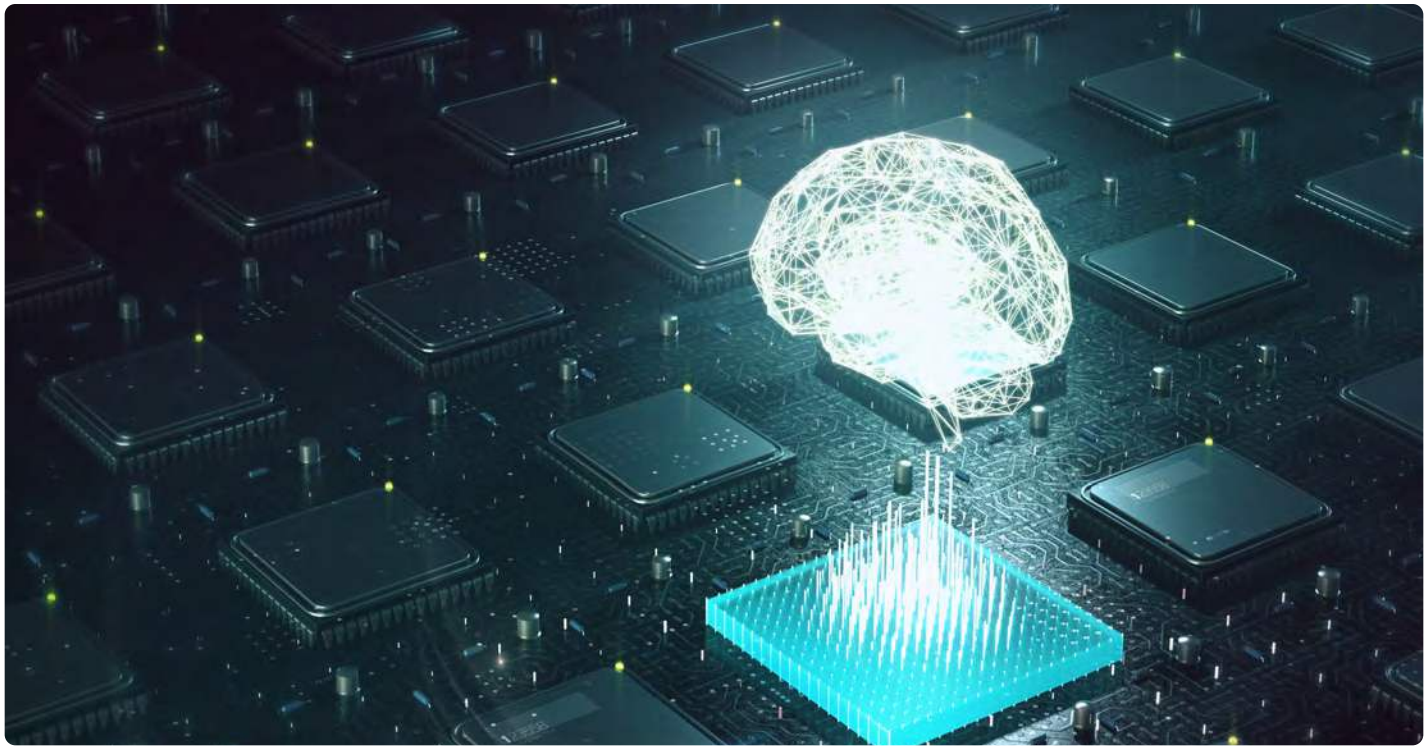
PREPARE FOR THE INEVITABLE

Generative AI and LLM tools are coming. The technology is already in place, so it's only a matter of time before they become a standard part of a DevOps toolbox. In fact, many developers are already using them. This inevitability means you need to start getting your systems ready to safely and properly address the potential downsides and get the most value from the benefits of these powerful tools.

Einstein GPT might be in the early stages of its rollout, but it's guaranteed to become part of your Salesforce DevOps approach.

This period of slowly rolling out tools and refining their capabilities is a great opportunity to get your systems in place. Anyone who isn't currently preparing to integrate these tools runs the risk of falling behind their competition.

Taking the time to plan this integration means that when the tools are fully integrated into your system, you'll have the processes in place to manage the increase in speed. Security and quality won't be as much of a concern if you are already scaled to meet the higher demands introduced by generative AI and LLM tools.



OFFER STRATEGIC TRAINING

We mentioned how manual errors are a constant concern for every DevOps process, but especially for AI tools because of their capacity for speed and volume. The first step to securing your processes around these tools is to provide ample training to your team members.

Taking a unified approach to generative AI and LLM tools helps team members understand how their actions impact projects as a whole. This reduces confusion and resulting errors, enabling team members to focus on refining their approach instead of putting out fires.

Expectations surrounding data privacy considerations must be clearly established. End-point protection, data anonymization, and consent management must all be built into the DevOps pipeline to address any sensitive information handled by your team as well as your AI tools.

Performing regular audits of your DevOps system will help administrators find potential data security vulnerabilities, identify improper usage of tools, and discover areas for improvement. There are bound to be difficulties when implementing these tools, so continued attention is necessary to correct any mistakes and provide guidance to educate team members and avoid making the same mistakes again.

DEVSECOPS TOOLS AS GUARDRAILS

If generative AI and LLM tools put rocket boosters on your DevOps projects, an automated DevSecOps suite will act as your safety guardrails. A non-negotiable aspect of this is a static code analysis tool. Automatically generated lines of code need to be tested as soon as possible to ensure any bugs don't sneak through additional layers of testing. This allows developers to rectify coding errors the moment they're written to streamline future processes.

Automated integration and deployment tools provide additional rounds of testing to ensure everything works properly when your code is integrated into the main repository. Speed and velocity are the main benefits of LLM tools, so it's up to your team to verify the reliability of the resulting code.

The unpredictability of AI tools—and DevOps in general—makes a frequent data backup snapshot essential to a complete security strategy. You simply never know what's going to happen, so you need to prepare for the worst. Frequent backup snapshots enable your team to quickly return to operations after an outage.

Thorough reviews of all third-party tools need to be conducted—including generative AI and LLM tools. Many of these are open-sourced, so they need to be consistently analyzed to see how they're addressing your system's data and security.

CONCLUSION

Advancements in software and tooling for Salesforce DevSecOps processes are always happening. The introduction of AI-enabled tools presents a massive shift in how teams approach their daily tasks.

Generative AI and LLM tools are in the early stages of introduction, but they are coming. Preparations need to be made as soon as possible to prepare your team to safely make use of them. Processes and supporting tools need to be in place when these tools are rolled out so your team can immediately harness the potential for increased speed.

Training your team is a critical component of this, but so is providing employees with the tools they need to verify proper structures of AI-generated lines of code. AutoRABIT's comprehensive DevSecOps suite offers everything your team needs to safely integrate AI tools into your DevOps processes.

Static code analysis, deployment automation, testing automation, and backup and recovery capabilities—these tools offer the guardrails you need to confidently utilize the capabilities of generative AI and LLM tools without risking compliance failure and data security vulnerabilities.

ABOUT AUTORABIT

AutoRABIT is a DevSecOps suite for SaaS platforms, which automates and accelerates the entire application development and release lifecycle. This enables continuous integration and delivery by providing fast, simple, and secure end-to-end automation across all Salesforce implementations. AutoRABIT tools help enterprises achieve higher release velocity and faster time to market.

AutoRABIT features static code analysis, automated metadata deployment, version control, advanced data loading, orgs and sandbox management, test automation, and reporting. Its services complement and extend Salesforce DX.

AutoRABIT Vault is a comprehensive backup and recovery solution that streamlines Salesforce data, simplifies data backup challenges, offers disaster recovery, and provides endpoint data protection in the cloud.

CodeScan gives Salesforce developers and administrators full visibility into code health from the first line written through final deployment into production, along with automated checks of Salesforce policies.

Record Migrator enables automatic bundling of all feature dependencies for Salesforce-managed packages. The deployment of templates ensures fast, efficient, and seamless releases.

Visit us at www.autorabit.com to learn more.

CERTIFIED
+ COMPLIANT

SOC 2 TYPE II CERTIFIED

CALIFORNIA CONSUMER PRIVACY ACT

GDPR

ISO 27001 INFORMATION SECURITY CERTIFIED

HIPAA COMPLIANT

> *"AutoRABIT has helped us add a lot of automation to our software development lifecycle. I highly recommend it!"*
>
> - FORREST COOK