SVT ROBOTICS

SOFTBOT® Platform

8 Key Considerations for Evaluating Automation Software

Introduction

Simplifying Automation Deployment Complexities

As global industries continue to struggle under the weight of supply chain challenges, a common solution is emerging: adaptable and flexible automation technology. Today, more companies are turning to automation to benefit from consistent and fast performance that can shore up supply chains and maximize productivity.

But knowing which solutions to deploy can be complex. Robots must be integrated into a company's existing environment — likely with an enterprise system, and with other automation technologies. (If not at first, then as business needs evolve.)

It's common for those hurdles to become significant, as a recent study from the Peerless Research Group for Modern Materials Handling magazine found:

"When adopting materials handling software, 39% of respondents say compatibility with existing systems is a major roadblock, while 38% point to integration with existing software applications. Other obstacles to implementation success include the total cost of ownership of such systems (37%); compatibility with host or legacy systems (36%); the lack of resources to implement, manage, and maintain such systems (35%); and substantiating ROI (30%)." [1]

"...39% of respondents say compatibility with existing systems is a major roadblock."

There are certainly big benefits to be had from automation. However, getting solutions—hardware and software — to communicate and work together in concert can be the barrier to entry you didn't see coming. Modern software platforms are tackling this interoperability issue and making it possible for companies to deploy custom configurations in flexible, secure, reliable, and low-code environments.

The answer is a technology-agnostic platform for integration.

Simple Integration Is Vital

In the not-too-distant past, deploying automation meant investing in an entire system. Many solutions in the industry tend to favor a single, comprehensive approach, often focusing on automating the most challenging aspect of the end user's workflow. This approach overlooks other important considerations and fails to address the diverse needs of different users and tasks.

But in the current rapidly changing market, automation flexibility is critical to future business viability. The same survey from MMH also found, as an example, that 45% of companies have had the same WMS in place for more than five years.[2]

As these systems age, they become a functionality liability and increase the challenges of deploying new technologies to the existing solution set.

That means contemporary solution deployments must include a simplified and flexible integration strategy that's easy to adapt as business needs change. Integrating complex software and hardware solutions across a facility or supply chain limits flexibility because the tech stack layers are nonstandard across all vendors. Even enabling data to pass between sub-systems is often black-boxed, causing IT and security conflicts. Additionally, as cloud-based applications become commonplace, upgrades are happening more frequently — along with their security requirements.

Collectively, these challenges present the scenario of maintaining your integrations against a constantly upgrading technology stack. This is extremely difficult but can be solved through technology-agnostic integration software.

[1,2] McCrea, B. (2022, July 12). Warehouse Software and Automation Survey 2022: Users cautiously embrace change. Modern Materials Handling.
https://www.mmh.com/article/warehouse_software_and_automation_survey_2022_users_cautiously_embrace_chan

Things to Evaluate

The latest top-tier software platforms enable integration with custom configurations without complex, application-specific coding. They are easily configured to meet today's operational challenges across a wide-ranging solution set. That said, the optimal software solution should include the following eight important technical aspects.

1. Modern Software Architecture

Software architecture refers to the underlying structure of the system. It has a significant impact on flexibility, security, and ease of modifications or updates, so evaluating the software's architecture is an important step. Typically, there are two types: monolithic and microservice.

Enterprise applications with monolithic architecture are built in three parts — a database, a client-side user interface, and a server-side application. All three are tied tightly together in a unified, single executable program. This approach is less scalable — with a large, complex code base that requires redeployment of the entire application with every update — resulting in slow development time and an infrequent release cycle.

In contrast, microservice architecture splits the software application into smaller, interconnected, self-contained services. Each microservice is a small application, controlling individual features connected through an application programming interface (API). This architecture enables each service to be developed and deployed independently, resulting in faster development, increased reliability, and more frequent releases.

2. Single Module Upgrades

Tasks in warehouse automation have been traditionally managed with business rules and branching logic. These complex workflow tools spawn a myriad of one-off, undocumented knowledge-based systems that contain unneeded scenarios. As a result, when business demands change at the rate they do today, the complexity of those proprietary systems becomes cumbersome and unmanageable.

Instead, the microservices on a technology-agnostic integration platform correspond to single module blocks strung together organically. While traditional workflow systems build everything from generic blocks that rely on a processing engine, modern platforms enable the construction of material flows without complex decision trees. Even when the need arises to build a new, more specific material flow, that adjustment process takes place only with the application in a single process block — not the whole system.

With the ability to easily add, revise, and replace microservice components, companies experience flexible supply chain management and marketplace readiness. Single module upgrades allow companies to combine different technologies to create novel solutions that better address their specific needs.

With the ability to easily add, revise, and replace microservice components, companies experience flexible supply chain management & marketplace readiness.

3. Security Approach

When selecting warehouse automation software, security is another vital area of evaluation. This includes the programming method used to prevent the application from being compromised, as well as how information is encrypted.

To prevent unauthorized intrusion into software systems, developers will deploy layers of infrastructure through APIs. One of the most secure software development methods is creating a virtual environment to run the application instead of running it directly on the operating system. Virtualization utilizes "containers" to restrict access to individual micro services. These containers host isolated micro services with individual firewalls to harden against system-wide access in case of an intrusion.

Another security consideration is the software developer's systems and organizational control. Companies face a growing threat to their security landscape, making information and data security a top priority. SOC 2 is a popular cybersecurity compliance framework that ensures third-party service providers store and process client data securely. Though voluntary, many companies employ the SOC 2 compliance standard, given it is a significant identifier that a software developer is committed to keeping their customers' data safe.

SOC 2 compliance is a significant identifier that a software developer is committed to keeping their customers' data safe.

4. Software Visibility and Reliability

In systems, every component is a potential point of failure, making reliability or problem-free operation critical to success. Since no software is entirely bug-free, evaluating the system's ability to detect, display, and then automatically recover from operational faults or errors is key.

Operational visibility is the cornerstone of reliable software. Logs and metrics are important for understanding and managing its functionality, and any software you consider should have those in place to allow monitoring and troubleshooting (ideally in one place to give operators a holistic view). That capability accelerates troubleshooting sessions and manual intervention because it's possible to simultaneously view and analyze the interactions.

Additionally, you'll want any software you purchase to automatically recover when it detects an operational fault, which will help reduce operator intervention. For example, if there is a data transmission error to a robot, you want a system that can simply retry sending the message.

Automated recovery in warehouse software improves reliability while reducing the impact of operational downtime. Software that lacks these capabilities requires more manual intervention and longer delays — both affecting cost and revenue.

5. Pre-built Integrations and Scalability

Today's software technology has radically changed the way automation solutions can be piloted and utilized, as the technologies are pre-built into the standard integrations of the platform. That means it's possible to test fully integrated robotics in the time it takes to deliver the hardware, and without time-consuming and expensive custom code.

Modern integration platforms should enable the pilot program and workflow results to reflect precisely how the technology will perform when integrated alongside other technologies in the existing warehouse. This real-time visibility enables companies to test a range of solutions — based on available edge technologies or different user interfaces.

6. Integration Across Multiple Solutions

As companies continue to increase the complexity of their workflows, it's paramount to have the ability to integrate and deploy multiple technologies. Traditionally, connecting automation technologies was done point-to-point; an integration process that involves custom development code to connect two technologies. However, point-to-point is only suitable when dealing with a straightforward integration.

As additional technologies are added to the solution set, point-to-point integrations become more time-consuming to implement and less future proof, forcing companies to depend on employees with specialized programming knowledge. (Made even more difficult in the current labor-strained market.)

In contrast, modern integration software platforms address the drawbacks of point-to-point by enabling rapid connection between technologies through pre-built connectors. Because the integrations are pre-built, these platforms offer low-code or no-code user interfaces, so companies no longer require a small army of software engineers to quickly scale integration.

Additionally, integration platforms centralize data visibility in one place. These dashboards enable teams to view the various activities performed across the solution set with customizable time frames.

Because the integrations are pre-built, these platforms offer low-code or no-code user interfaces.

7. Support and Third-Party Development Capabilities

In addition to hardware, supply chain integration is increasingly focused on connecting software. These integrations serve the purpose of creating a connected experience, allowing companies to move data back and forth between systems. Software product integrations are designed for customers to be able to work seamlessly with all the products and services they utilize without having to manually move data or continuously toggle between systems. That is accomplished through either native or third-party integrations.

Native integrations are built directly between two or more developers, creating a rigid system where companies are unable to expand and connect additional apps to meet their evolving business needs. Conversely, third-party integrations standardize the flow of data between multiple applications, allowing the possibility of connecting hundreds of apps. When considering different software options, look for a vendor that supports and develops third-party integrations for maximum flexibility.

8. Hosting Methodology

As cloud computing and broadband capabilities have increased, so have the hosting options. Before the cloud, on-premises storage was the only option that enabled companies to set up and manage their own servers to host data and applications. As adoption is becoming more popular, many software vendors are now offering their applications in the cloud through a software-as-a-service (SaaS) model.

The SaaS, or cloud model, has many benefits, including lower up-front costs and fast implementation without having to upgrade or maintain on-premises hosting infrastructure. Plus, the application is available anywhere with internet access. However, SaaS applications tend to offer little in terms of customization options and can be vulnerable to reliability issues because clients have less control over the system.

In contrast, deploying applications to on-premises servers maintained in the facility, offers many benefits, such as numerous customization options, client-side security, offline access, and reduced latency. Additionally, implementation, hardware upgrades, and software updates are controllable, increasing reliability because the software is deployed only for a single client.

Both methods are viable options when considering software. Companies must consider the cost, risk, and customization of each. However, top-tier integration software platforms will leverage the benefits of on-premises and cloud applications for the greatest usability.

Conclusion

Simple integration via best-in-class software offers flexibility, reliability, and reduced deployment times. Software that enables companies to respond rapidly to supply chain demands should address the following:

- **Modern Software Architecture** — Solutions with microservice architecture enable self-contained applications to control individual features. This provides faster development, increased reliability, and frequent release cycles.
- **Single Module Upgrades** — Modern software platforms facilitate the construction of material flows without complex decision trees that allow the replacement of microservice components for flexible supply chain management.
- **Security Approach** — Software that leverages multiple layers of security, including firewalls, virtualization, and containers, reduces the risk of data breaches. View security holistically and evaluate the developer's system and organizational control.
- **Software Visibility and Reliability** — Reduce the risk of downtime with software that offers operational visibility for monitoring and troubleshooting. Software that can automatically recover from operational faults reduces operator intervention for greater reliability.
- **Pre-Built Integrations and Scalability** — Modern integration platforms should enable the pilot program and workflow results to reflect precisely how the technology will perform when integrated alongside other technologies in the existing warehouse.
- **Integration Across Multiple Solutions** — Mitigate the drawbacks of point-to-point integration by enabling rapid connection between technologies using pre-built connectors (therefore no- or low-code user interfaces) across the entire solution set.
- **Support and Third-Party Development Capabilities** — Search for a vendor that supports and develops third-party integrations. These standardize the flow of data between multiple applications, allowing for maximum flexibility.
- **Hosting Methodology** — Both on-premises and cloud hosting are viable options. Top-tier software leverages the benefits of both, for easy implementation, increased reliability, and optimization.

Once automation technologies are connected to the platform, end users can simply select the best solution for their needs.

By choosing the right software, companies can address interoperability challenges and make it possible to deploy custom configurations in flexible, secure, reliable, and low-code environments. Using a technology-agnostic platform for integration means that once technologies are connected to the platform, end users can simply select an automation solution and then integrate it with their enterprise system in just days or weeks — all without tasking software engineers with extensive coding.

SVT Robotics' technology-agnostic SOFTBOT® Platform comprises a configurable, pre-built integration layer that enables rapid and simplified automation deployment between enterprise systems and multiple automation technologies. Utilizing our extensive warehouse automation and industry knowledge, we've pre-built the tasks and workflows we know many companies need for today's supply chain.

Want to learn more about how the SOFTBOT Platform can work for you? SVT's experts are ready to help. Contact us at **contact@svtrobotics.com**.