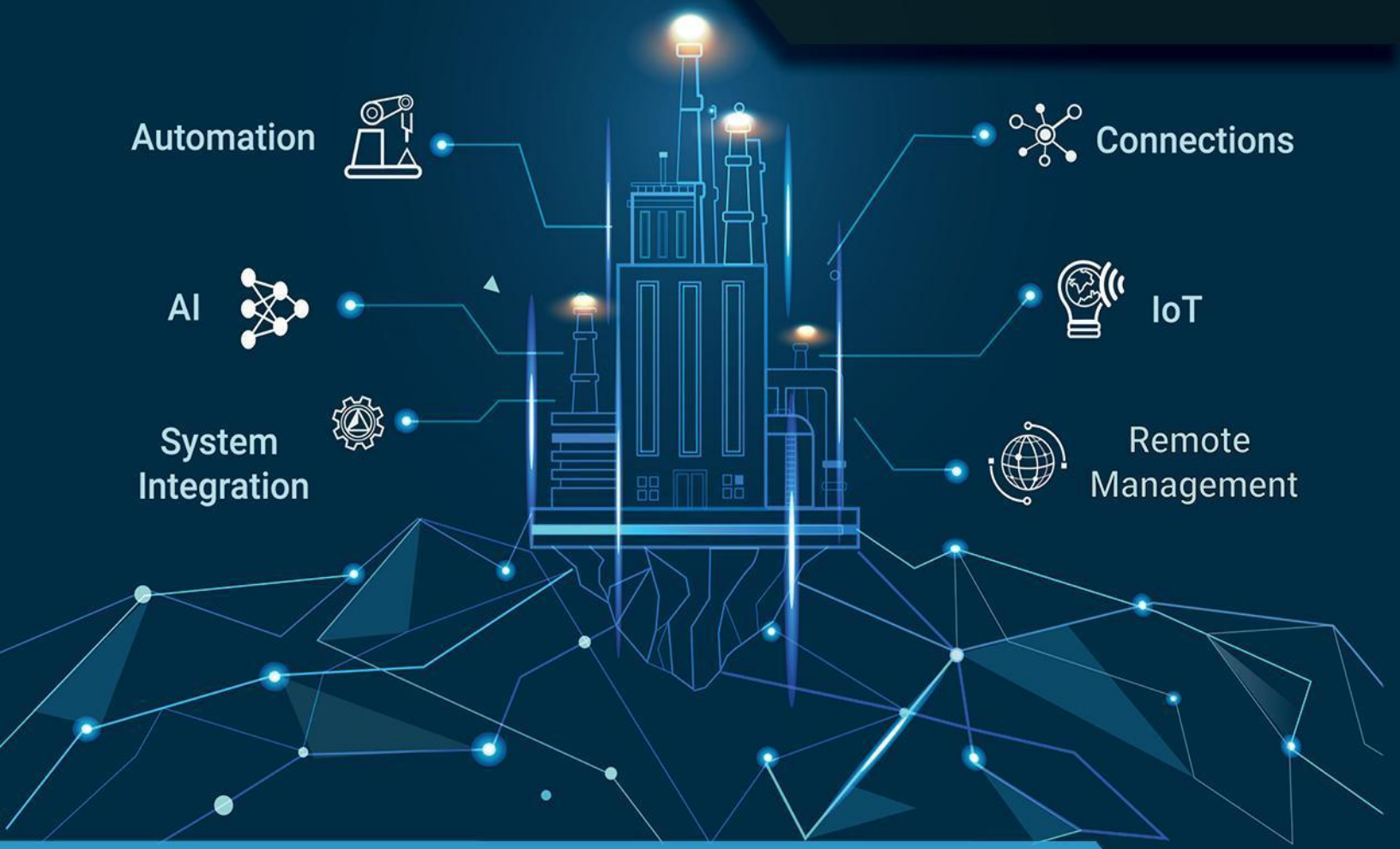
WHITE PAPER



Accelerating Smart System Software Development

What Every CTO and Engineering Leader Should Know

numurus
SMART SYSTEM SOLUTIONS

Accelerating Smart System Software Development

What Every CTO and Engineering Leader Should Know

OVERVIEW

Industry 4.0 is the ongoing transformation of traditional industrial practices to more automated and interconnected operations. This transformation is underway and is quickly changing the face of many businesses and leaving those that fail to adapt irrelevant.[1] Graham Immerman, in his article, "Industry 4.0 Advantages and Disadvantages," provides the following benefits which are driving investments in Industry 4.0 technologies and solutions: competitive advantage, increase in operational efficiency, better products and services, growth of markets or new markets, improving lives overall.

While many solution provider companies invest in developing industry 4.0 solutions there is a realization of how complex creating such solutions is and the painstaking effort, yet necessary, of orchestrating many new and rapidly developing technologies – all integrated into a seamless, complete product solution.

As end-customers plan their industry 4.0 system upgrades, this paper provides CTOs and other leadership of smart sensors, robotics and Edge-AI solution providers some helpful insights about the challenges of building smart system software in-house. Equally as important, this paper covers the important considerations that must be taken into account as you evaluate buying vs. building the underpinning software that is one of the single most important factors in attaining industry 4.0 product line capabilities.

[1] Machine Metrics, "Industry 4.0 Advantages and Disadvantages," Graham Immerman, February 2018, Link

CHALLENGES

There are significant challenges that should be considered as you embark upon a smart sensor, robotic or Edge-AI smart system development.

Disparate Complex Technologies

Modern smart systems are possible because of an underlying layer of disparate complex software technologies that each require software developers with deep, specialized knowledge and experience. As Anders Beck states in his article 5 Robotics Predictions for 2023, while companies are driving innovations by combining these technologies, all companies suffer from the same problem – they can't be good at everything.

Lack of Specialized Engineering Labor

With nearly every industrial enterprise either looking to modernize their existing products or adopt completely new smart solutions, solution providers are faced with a serious shortage of engineers who can adequately work with these technologies.[2] Software engineer positions in general have been difficult to fill for the last several decades, and engineers with highly specialized skill sets – the very skill sets needed for industry 4.0 transformation – are even more difficult to find.

Development Costs

The development of smart systems requires a lot of low-level software adding a significant cost burden for such efforts. The high costs and challenges of hiring or contracting specialized engineers that can support this work are major hurdles in getting such projects off the ground. In addition, protracted technical problems during the development will significantly erode profits for smart systems by delaying launch dates and incurring additional engineering expenses.

Development Time

Technology innovation is moving at a rapid speed. Market forces are pushing end-customers to adopt new technologies at a rapid clip demanding solution providers to adopt rapid innovation. Technology development companies are pressured to build smart products faster due to a plethora of influences. Completing projects on time is hampered by the fact that building full-stack software from the ground up requires significant capital investments in non-core technologies and can take years to complete.

Maintenance and Support

With engineers specialized in industry 4.0 technologies at a premium, you must consider the cost and time required to maintain smart system software and the potential distraction this effort could have on your core business including the time and costs of continuously introducing new features after a baseline product is released.

[2] Robotics & Automation, "Software that is Helping the Development of Robotics," David Edwards, November 2021, Link

CONSIDERATIONS

With every new project, you must make decisions on where to spend your limited resources. Build vs. buy decisions are now taking center stage. The debate is nothing new and is a common decision that must be made at the start of each new development project. Nabeel Wyne in his article, "Build vs. Buy Debate," states, "the goal of this decision is to find an available platform product from the market that requires only limited customization for a specific solution, as opposed to the build path which refers to writing custom applications from scratch using an internal development team." [3]

There are several factors to be considered in your build vs. buy analysis:

Purpose: Does the software serve a core function of the business, or is it non-core development to your solution? If it serves a core function, it may make more sense to build the software in-house to ensure that it aligns with the company's specific needs and processes. On the other hand, if it is non-core development it may be more cost-effective to buy an off-the-shelf solution.

Expertise: Does the company have the necessary expertise and resources to build and maintain the software in-house? Building software requires specialized skills and resources, so it may not be feasible if the company does not have the necessary expertise or capacity.

Time and Cost: Building software can be time-consuming and costly, especially if it requires significant customization. You must consider the full product life-cycle maintenance overhead each new in-house developed software application will require each year, versus the value of having that maintenance component provided by a third-party platform product.

Flexibility: Do you need a highly customized solution that can be tailored to its specific needs and processes? If so, building the software in-house may be the best option. On the other hand, if you are looking for a more generic solution that meets most of your company's needs, buying an off-the-shelf product may be a more cost-effective option.

Full Scale's article, "Build vs. Buy Software, Which is Better?" [4] provides a clear set of criteria for buying versus building software in-house:

- **The software is not a core part of the business.**
- **You're on a budget and you need a cost-efficient tool.**
- **You're on a schedule and you're aiming for fast deployment.**
- **You're not equipped with the technical skills and knowledge required.**

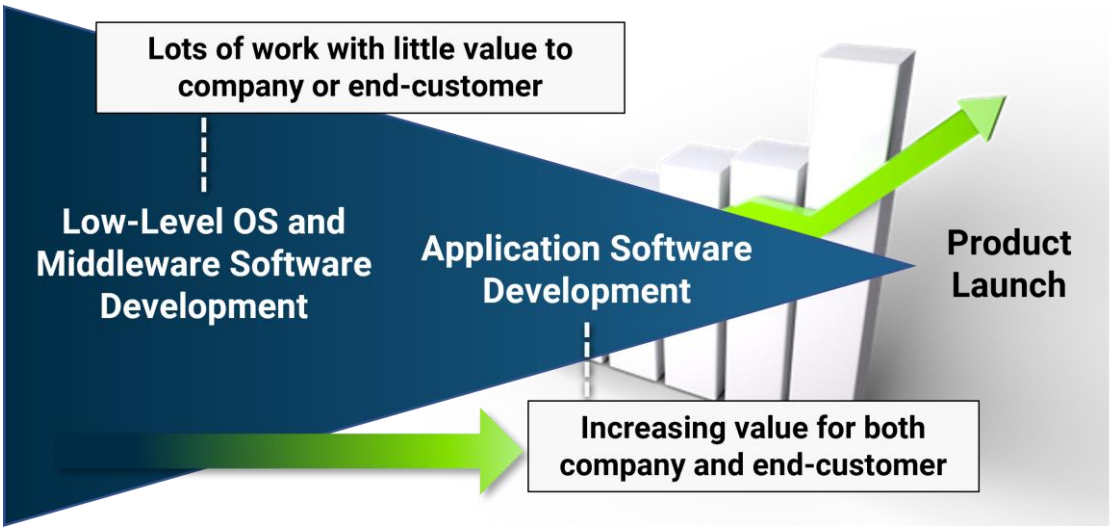
Ultimately, your decision to buy or build software should be based on a thorough analysis of the costs, time and resources required and your company's specific needs.

[3] LinkedIn,"Build vs. Buy Debate," Nabeel Wyne, April 2021, Link
[4] Full Scale, "Build vs. Buy Software, Which is Better?" August 2021, Link

ACCELERATION

A key factor in accelerating smart system software development is determining which software components are best built in-house versus leveraging commercial software development platforms. A complete smart system product solution is the combination of disparate software components, and while some provide core product differentiating capabilities like navigation, planning, decision making, and other application layer features, most of the software required is low-level nuts and bolts, the operating system and specific open-source packages such as the Robot Operating System (ROS) library, which provide little to no strategic value for a company but consumes extensive resources building everything from scratch.

Smart systems include extensive interfacing, and testing a mix of GPU-optimized operating system software with sensor, data, AI, event, IoT, communications, and other system management solutions. All of these low-level components must first be in place and matured before development teams can start working on their core application-level software, and each of these underlying technologies requires software developers with specialized knowledge and skills. In contrast, the application layer software is where nearly all of a company's intellectual property (IP) and value to the end customer comes in, and should be the main area of focus for any in-house development effort.

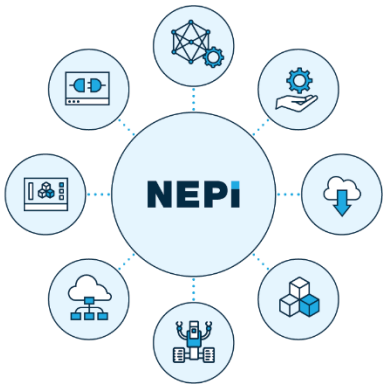


The low-level operating system and middleware components of a smart system solution are extensive, yet as the figure above demonstrates, it adds little value to the company as in-house built solutions and increases development costs and time significantly. In Christian Fritz's article, "A Full-stack Platform for Robotic Applications," Fritz suggests very few of these capabilities provide differentiation for the companies who build and use them, yet each robotics startup independently recreates these capabilities in-house, reinventing the wheel.[5] Based on this analysis of the smart system software development process, it is clear that off-loading low-level

[5] Towards ROSboics, "A Full-stack Platform for Robotic Applications", Christian Fritz, February 2021, Link

software components such as operating systems and middleware are an area in which organizations can accelerate their smart system software development.

NEPI SMART SYSTEM SOFTWARE



Until recently, low-level and middleware software components were not available as a complete and bundled commercial-off-the-shelf or open-source solution that robotic and smart sensing companies could buy and build off of, requiring each company to independently build and maintain these software components in-house, wasting precious resources on low-value and time-consuming development efforts. As Christian states at the end of his article, and which presumably many CTOs and company leadership have thought, “there’s gotta be a better way.”

Luckily, Numurus LLC provides a commercially available ‘buy option’ that solve this dilemma with its NEPI smart system software, a turnkey, open-source Linux software distribution for robotic, smart sensing and Edge-AI applications that integrates all the required low-level and middleware disparate technologies into one commercial offering along with regular feature and security updates. Developers leveraging NEPI software have access to a full support team that includes AI, robotic, and smart sensing specialized engineers and developers, and can avoid wasting days or weeks struggling to find software issue resolutions on outdated open-source forums. With NEPI, companies can reduce the time, cost, and risks to develop smart systems and free-up engineers to focus on their end-customer applications.

CONCLUSION

There are many considerations to take into account as you continue to modernize your products. Integrating diverse technologies required to support your goals is a major effort but one where there is now help. Doing this work in-house requires engineers and software developers with deep, specialized knowledge and experience in a wide variety of cutting-edge technologies. Most companies face challenges staffing such a team and justifying the value of developing non-core software in-house. Unless building low-level software is a core component of your business, you are much better off investing in commercial development platforms, such as NEPI, that can reduce development efforts, eliminate the need to staff expertise across all the smart technologies and significantly increase your speed to market.

ABOUT THE AUTHOR

Jason Seawall, CEO at Numurus LLC

Jason Seawall has over 20 years of experience in the field of robotics and smart sensing. Jason was a research engineer at both the Applied Research Laboratory in Texas and the Applied Physics Laboratory in Washington working on robotic sensors for defense applications. He founded and sold BlueView, a marine robotic sensor company to Teledyne in 2012 and took the role of VP of Technology in Teledyne's marine robotics and sensing division. In 2017, Jason co-founded Numurus to focus on products that accelerate the development of autonomous robotic systems.

CONTACT

Want to learn more about NEPI Smart System Software? Please contact us:

info@numurus.com

www.numurus.com

Linkedin/numurus