

VORSICHT VOR EXTREMER SOFTWARE



Es herrscht die weitverbreitete Auffassung, dass ein guter Mittelweg darin besteht, Kompromisse einzugehen. So werden angeblich die meisten Menschen zufriedengestellt. Allerdings ist oftmals das genaue Gegenteil der Fall.

Wie groß die Enttäuschung ist, hängt davon ab, wie weit man hinter den Erwartungen zurückbleibt und wie der kleinste gemeinsame Nenner lautet. Gilt dieses Prinzip auch für die Herstellung von Software? **Sind Extreme eine Lösung, um Kompromisse zu vermeiden oder gibt es einen ausgewogenen Ansatz, mit dem wir die digitale Transformation vollständig umsetzen können?**

Verhandlungen um eine bestimmte Situation sind oft schwierig, denn jeder von uns hat unterschiedliche Erfahrungen, Ansichten, Erwartungen und andere Prioritäten. Um einen kompromisslosen Konsens zu erreichen, muss man zunächst die wahren Bedürfnisse, Wünsche und Vorlieben jeder Partei kennen und dann die Ursachen und potenziellen Folgen der einzelnen Standpunkte begreifen. Extreme Standpunkte sind meist nur einseitig und lassen andere Faktoren und Überlegungen nicht gelten. Dabei ist es äußerst wichtig, alle Standpunkte genau zu analysieren, auch wenn das mühsam ist.

Einzelne Elemente in der Softwareentwicklung und Lösungspakete haben zahlreiche unterschiedliche und manchmal extreme Standpunkte ausgelöst, die alle einen Einfluss auf die Softwarelösungen, ihre Kernfunktionen und ihre letztendliche Nachhaltigkeit haben. Der Käufer einer Softwarelösung muss mit Anbietern verhandeln können und nicht nur die hervorgehobenen Vorteile der Lösung erkennen, sondern auch deren Betriebskosten und nicht vorhandene Möglichkeiten

Zwei grundlegende Bereiche von entscheidender Bedeutung, die bei extremen Lösungen fast immer übersehen werden, betreffen eine definierte Ontologie und Sicherheit. Die Ontologie in der Software stellt die Mechanismen dar, die hinter der Datenstruktur und Modellierung von Informationsregeln stehen, und die den Kern der Wertschöpfung ausmachen. Fehlt sie, kann jede Extremlösung Daten erfassen und interessant darstellen, schafft allerdings kaum einen Mehrwert.

Bei der digitalen Transformation geht es nicht nur um automatisierte Datenerfassung und Berichtswesen. Aus Sicherheitsperspektive erfassen viele Extremösungen sämtliche Daten und schaffen "Big Data"-Repositorien, ohne die Sicherheit der Inhalte des geistigen Eigentums oder den Schutz wichtiger Informationen zu beachten. Das Ergebnis ist dann "freier Zugang für Alle" in der Hoffnung, dass kein Unbefugter die Daten stiehlt, verändert, oder Schadsoftware einführt.

Sämtliche Lösungen basieren auf individuellen Philosophien, die sich im Extremfall sehr stark unterscheiden. Allerdings sind Faktoren wie Mehrwert, Kosten und Risiken jedoch sehr ähnlich. Alle extremen Ansichten halten ihren Ansatz für den einzig richtigen und schaffen ein unausgewogenes Lösungsmodell, das, ohne dass dies den potenziellen Nutzern bewusst ist, leicht zu einem erheblichen Verlust an Möglichkeiten führen kann. **Ein Blick auf diese extremen Ansätze verrät viel über die langfristige Lebensfähigkeit und Nachhaltigkeit einer Softwarelösung.** In diesem Whitepaper werden einige Beispiele für die häufig erlebten Extreme solcher Philosophien in Bezug auf MES-Software betrachtet

Inhaltsverzeichnis

- 2 Einleitung
- 3 Extrem: In-House Entwicklung
- 5 Extrem: Open-Source
- 6 Extrem: Dominanz kommerzieller Lösungen
- 7 Extrem: App-basierte Technologie
- 10 Extrem: Interoperabilität
- 12 Extrem: Flexibilität
- 14 Fazit

Extrem: In-House Entwicklung

Software, die intern entwickelt wird – von komplexen Lösungen bis hin zu kundenspezifischer Middleware – folgt einem “Do-it-yourself“-Ansatz, der noch heute relevant ist. Derartige Lösungen sind eng umrissen und auf aktuelle Bedürfnisse ausgerichtet und werden von einem internen Entwicklungsteam und/oder einem ausgelagerten Partner entwickelt.

Es gibt eine klare und eindeutige Spezifikation, die, einmal vereinbart, wenig Flexibilität erfordert. Die Software selbst kann verkleinert und vereinfacht werden, und die Architektur und das Datenmodell können so gestaltet werden, wie es für das jeweilige Projekt am besten geeignet ist. Der Softwareentwickler hat das Gefühl, die volle Kontrolle zu haben, und die Benutzer haben die Gewissheit, dass die Anwendung so funktioniert, wie es für ihre spezifischen Aufgaben am besten ist. Die Vorteile sind leicht messbar und die Lösung wird schnell zur Gewohnheit. So lautet die gemeinsame Geschichte tausender “intern entwickelter“ oder “hausgemachter“ Punktlösungen, die es heutzutage in der Fertigung gibt, jede von ihnen kann als ein Element der MES-Funktionalität betrachtet werden. Kleinere Hersteller sind oftmals der Auffassung, dass dieser Ansatz ein risikoarmer Startpunkt für ihre digitale Transformation ist. Größere Unternehmen bevorzugen es, diese Technologie fest unter ihrer eigenen Kontrolle zu haben.

Kleinere Hersteller sind oftmals der Auffassung, dass dieser Ansatz ein risikoarmer Startpunkt für ihre digitale Transformation ist. Größere Unternehmen bevorzugen es, diese Technologie fest unter ihrer eigenen Kontrolle zu haben.

Da die Vorteile dieser Lösungen immer stark betont werden, werden die vielen Herausforderungen, die diese Lösungen mit sich bringen, oftmals übersehen. **Bei extremen intern entwickelten Lösungen wird die Spezifikation von wenigen internen Architekten übernommen.** Es gibt ein oder zwei Fachleute, die Sie unterstützen können, das Ergebnis ist jedoch ein sehr enges und vereinfachtes Datenmodell beziehungsweise Ontologie. Es sollen lediglich die aktuell definierten Werte aus den Daten herausgeholt werden, wobei andere potenzielle Werte nicht berücksichtigt werden.

Wird die Software von einem internen Team entwickelt, wird dieses Team sich im Laufe der Zeit verändern. Das Know-how über das Datenmodell, die Algorithmen und die Methoden der Software gehen in der Regel verloren, so dass es für andere schwierig wird, die Software weiterzuentwickeln und zu pflegen, was oft Reverse Engineering erfordert. Dabei kommt es häufig zu falschen Annahmen und es fehlt eine professionelle Dokumentation.



Die größten Herausforderungen bei extremer In-House-Entwicklung



Enges Datenmodell



Vereinfachte Lösungsarchitektur



Im Laufe der Zeit Verlust von Fachwissen



Schwierigkeiten bei fortlaufender Entwicklung

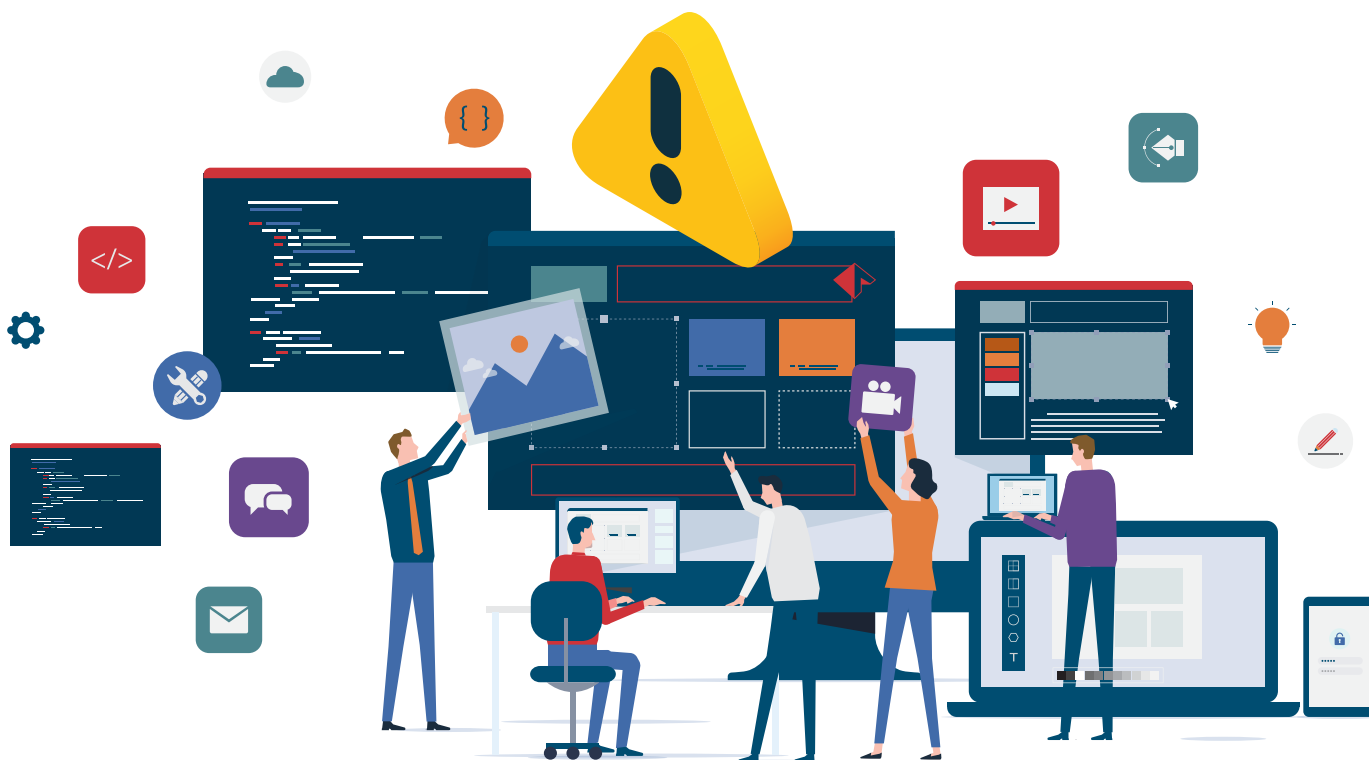


Hohe Kosten für Modifikationen

Werden solche Lösungen von ausgelagerten Partnern in Form von kundenspezifischen Anpassungen entwickelt, treten dieselben Probleme auf. Jedem Projekt sind Teams zugeordnet, die im Falle von Änderungen oder Ergänzungen wahrscheinlich nicht dieselben sein werden. Spezifisches Know-how bleibt nur in der Dokumentation und Spezifikation erhalten. Dies führt zu weitaus höheren Kosten als erwartet, sollten zu irgendeinem Zeitpunkt Änderungen oder Wartungsarbeiten erforderlich sein.

Die vereinfachte Lösungsarchitektur wird zu einem großen Hindernis für Erweiterungen oder weitere Innovation. Der Bedarf an Konnektivität zwischen derartigen Punktlösungen und anderen Automatisierungen einschließlich der Notwendigkeit von Cybersicherheit, fehlt oft völlig. Lösungen, die auf extremer Eigenentwicklung beruhen, sind daher schnell veraltet und schränken den Fertigungsbetrieb auf veraltete Praktiken ein, was eine echte digitale Transformation im Vergleich zu anderen Unternehmen der Branche verhindert. Es entsteht eine zunehmende Abhängigkeit von Schlüsselpersonen, die immer schwieriger zu halten sind, bis solche Lösungen schließlich "unantastbar" werden. Niemand, der im Team verbleibt, traut sich noch zu, das System im Einklang mit Betriebssystem-Upgrades und der Anwendung von Sicherheitsmaßnahmen zu warten, geschweige denn Änderungen vorzunehmen.

Es kommt häufig vor, dass interne Spezialisten für derartige Lösungen, die ihr Gesicht wahren wollen, die gleichen Projekte immer und immer wieder wiederholen, wobei die Kosten schneller steigen als neue Vorteile entstehen. Es braucht einen radikalen Wandel auf allen Ebenen eines Unternehmens, um diesen Kreislauf und diese Abhängigkeiten zu durchbrechen und in die Zukunft zu starten.



Extrem: Open Source

Ein weiterer extremer Ansatz in der Entwicklung von Software, sei es als Teil einer selbst entwickelten Lösung oder zum Füllen von Lücken in der Funktionalität und Konnektivität zwischen bestehenden Punktlösungen, ist die Verwendung von Open-Source-Softwarekomponenten. Der Gedanke, die Entwicklungskosten auf mehrere Entwickler zu verteilen erscheint mit dem Gefühl, etwas umsonst zu bekommen, verlockend.

Open-Source-Software ist in der Regel kostenlos und oft ohne restriktive Lizenzierung erhältlich. Softwarebibliotheken, Entwicklungskits (SDKs) und Codemodule können bezogen und in firmeninterne Lösungen integriert werden, um die Belastung der Entwicklungsteams zu verringern. Vertrauenswürdige Open-Source-Software, die zum Beispiel von einer zuverlässigen, auf einem Branchenkonsens basierenden Organisation für einen bestimmten Zweck bereitgestellt wird, kann sehr nützlich sein. In der überwiegenden Mehrheit der Fälle gibt es jedoch erhebliche Sicherheits- und Integrationsprobleme.





Obwohl es viele Open-Source-Projekte gibt, ist fraglich, ob ein wesentlicher Teil einer MES-Lösung zusammengestellt werden kann. Nur sehr wenige Open-Source-Teilnehmer teilen wirklich wertvolles geistiges Eigentum (IP). **Open-Source Komponenten sind eher Enabler als Lösungen.** Das Vertrauen in die Herkunft eines solchen Open-Source-Codes ist mit Vorsicht zu genießen, da wahrscheinlich viele Entwickler mit unterschiedlichen Hintergründen und möglicherweise widersprüchlichen Motiven daran beteiligt waren. Dies kann sich auf die Qualität des Codes auswirken, der möglicherweise auf mehreren Ebenen verkompliziert wurde, da er so manipuliert wurde, dass er in verschiedenen Anwendungsfällen unterschiedlich funktioniert.

Das führt dazu, dass Open-Source-Software viel größer und komplexer ist als es eigentlich nötig wäre. Es erschwert die Überprüfung solcher Software (z. B. zur Beseitigung potenzieller Schadsoftware oder anderer unerwünschter Effekte) erheblich, wobei eine zeilenweise Forensik erforderlich ist, wenn Open-Source-Code in einer geschäftskritischen Umgebung eingesetzt werden soll.

Die potenziellen Nachteile in Bezug auf die Sicherheit und das Verständnis des vollständigen Codes sowie fehlende spezifische MES-orientierten Funktionen machen die Einbeziehung von Open-Source-Software von nicht vertrauenswürdigen Organisationen unpraktisch und gefährlich. Der verbindende Code, der benötigt wird, um unterschiedliche Open-Source-Module zu verbinden und eine robuste Geschäftslogik und Datenmodellierung zu unterstützen, birgt immer mehr Probleme und Kompromisse, die direkt auf inhärente Einschränkungen zurückzuführen sind.

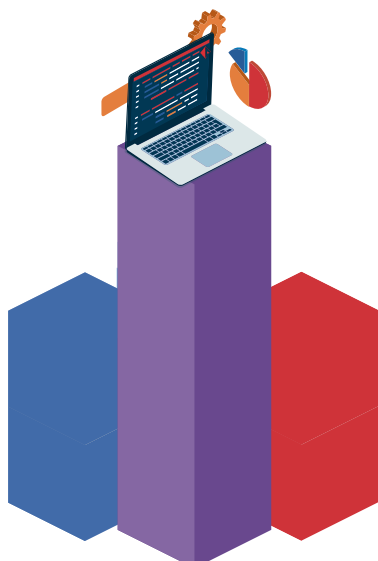


Die wichtigsten Herausforderungen bei extremer Open-Source

-  Herausforderungen bei der Sicherheit
-  Herausforderungen bei der Integration
-  Schlechte Code-Qualität
-  Aufblähung und übermäßige Komplexität schlechte Code-Qualität



Extrem: Dominanz kommerzieller Lösungen



Marktdominanz ist eine Geschäftsdoktrin, die auf Gewinnmaximierung basiert und darauf abzielt, Wettbewerb und Interoperabilität zu unterbinden. Diese Strategie kann bei vielen Lösungsanbietern auf dem Markt beobachtet werden: von den größten und teuersten bis hin zu den einfachsten App-basierten Umgebungen.

In der Natur wird der stärkste Nestling versuchen, seine Geschwister wegzudrängen, um das meiste Futter zu bekommen. Unternehmen wenden dieses Prinzip mit ihren Lösungen auch auf Kunden an, um diese zu dominieren und alle verfügbaren Aufträge zu gewinnen. Wenn ein solcher MES-Anbieter von einem Kunden dazu aufgefordert wird, Angaben zu den von ihm angebotenen Funktionen und Merkmalen zu machen, z. B. anhand einer Checkliste mit fünf benötigten Schlüsselmerkmalen, werden die dominierenden Unternehmen immer versuchen, jedes Kästchen abzuhaken, unabhängig davon, ob ihre vorhandenen Funktionen wirklich den angegebenen Anforderungen entsprechen.

In der Realität können Hersteller vielleicht drei der fünf Anforderungen erfüllen, die vierte Anforderung wird teilweise unterstützt, da sie vielleicht erst vor Kurzem für einen anderen Kunden entwickelt wurde. Die fünfte Anforderung wird wahrscheinlich gar nicht unterstützt, aber der Hersteller versichert, hierfür eine Lösung für den Kunden zu entwickeln. Ziel solcher Aussagen ist es, dass der Kunde mit dieser Lösung zufriedener ist als mit einer Lösung von der Stange eines Konkurrenten, da sie ja speziell für ihn entwickelt wurde.

Die Anbieter stellen ihre Schwäche also erfolgreich als Stärke dar. **Das Ergebnis ist jedoch eine kundenspezifische, maßgeschneiderte Lösung, die in Bezug auf Struktur, Mehrwert und Risiko einer extremen, intern entwickelten Lösung ähnelt.** Verzögerungen bei der Einführung und die damit verbundenen zusätzlichen Kosten werden oft übersehen, da dieser Anbieter als einziger "alle Kriterien erfüllt".




In all diesen Fällen gibt es die wesentliche Einschränkung, dass der Kunde nur Praktiken und Betriebsabläufe ausführen kann, die von der Lösung vorgegeben werden. Es fehlt die Flexibilität, Lösungen anderer Parteien oder Anbieter zu verbinden. Alles außerhalb der bestehenden Optionen kann nur durch kostspielige kundenspezifische Anpassungen nachgerüstet werden. Auf diese Weise werden maßgeschneiderte Punktlösungen geschaffen, die außerhalb regelmäßiger Support- und Wartungsverträge existieren. Bei jeder Aktualisierung des Kernsystems müssen solche maßgeschneiderten Ergänzungen überprüft und möglicherweise neu entwickelt werden.

Ein weiterer Aspekt einer dominierenden Lösung ist, dass der gesamte digitale Transformationsprozess dann durch das Angebot des einzelnen Anbieters eingeschränkt wird.

Für einfache App-basierte Plattformen sind das Lösungen mit dem kleinsten gemeinsamen Nenner an Funktionalität. Bei umfangreichen Lösungen wird sich die Nutzung zusätzlicher Funktionen wahrscheinlich nicht lohnen, da solche Lösungen häufig viel komplexer sind als nötig. Die Best Practices in der Branche und Technologien zur digitalen Transformation entwickeln sich rasant weiter und lassen die Nutzer dieser dominanten Lösungen oft in der Schwebel, da sie sich in einem Tempo und auf eine Art und Weise weiterentwickeln, die für ihr Unternehmen möglicherweise nicht geeignet ist.



Die größten Herausforderungen bei extremer Lösungsdominanz

-  Maßgeschneiderte Lösung
-  Starre betriebliche Abläufe
-  Hohe Anpassungskosten
-  Beschränkung der Innovation durch den Anbieter
-  Hoher Grad an Komplexität

Extreme: App-basierte Technologie

Seit vielen Jahren wird immer wieder behauptet, dass Softwareentwickler bald nicht mehr gebraucht werden. Softwarelösungen könnten erstellt werden, ohne dass dafür Code entwickelt werden muss. Doch diese Aussage ist falsch und, bis auf einige wenige Ausnahmen, Unsinn

Es gibt viele Lücken, in denen spezifische Kundenanforderungen unerfüllt bleiben, z. B. in Bezug auf die Geschäftslogik, die Einbindung der neuesten KI-basierten Algorithmen, Maschinen- und Geräteschnittstellen und vieles mehr.

Der Gedanke hinter aktuellen App-basierten MES-Lösungen ist, dass Bibliotheken den gesamten notwendigen Code enthalten, der die benutzerbasierte Erstellung unterstützt. Die Apps können jederzeit geändert werden, ohne dass der Kerncode in den Bibliotheken geändert werden muss

Es gibt immer wieder Schlagzeilen, die diese "no-code", app-basierte Idee stützen. Viele in der Fertigungsindustrie haben schon erlebt oder gehört, dass dominante MES-Lösungen eingeführt wurden, die alles versprochen und nichts hielten. Jahrelange Bemühungen und Anpassungen durch den Anbieter, den Kunden und Drittanbieter von Middleware forderten ihren Tribut, mit hohen Kosten, vielen Bereitstellungsproblemen und mangelnder kohärenter Unterstützung für die selbst entwickelten Anpassungen. **Die Versprechungen einer Software-Umgebung klingen verlockend, jeden Tag eine App erstellen zu können, um sämtliche Anforderungen zu erfüllen.** App-basierte Plattformen beinhalten häufig Open-Source-Communities, die Ideen und Programmierkenntnisse austauschen, um Kunden bei der Konfiguration ihrer Anwendungen zu unterstützen.

App-basierte Lösungen verwenden Bibliotheken, die MES-Anwendungs-/lösungsspezifischen Code auf hoher Ebene enthalten, der schnell veraltet, insbesondere wenn sich solche Lösungen aus einfachen Ansätzen entwickeln. Ironischerweise weisen die heutigen App-basierten MES-Lösungen dieselben Risiken und Herausforderungen auf, die auch bei extremer Inhouse-Softwareentwicklung, extremer Open-Source-Entwicklung und bei kommerziellen Lösungen, die den Markt dominieren wollen, auftreten. **Sämtliche Änderungen innerhalb der App-basierten Anwendungsbibliotheken bergen das Risiko, dass die gesamte Schicht der bestehenden Apps, die von den Kunden erstellt und konfiguriert wurden, sowie alle Methoden, der Code und die Funktionen, die von der Support-Community bereitgestellt wurden, beeinträchtigt und ungültig werden.** Die Verantwortung, dass die Lösungen funktionieren, liegt beim Kunden und die Anwendungen verhalten sich oftmals unvorhersehbar. Darüber hinaus gibt es viele Lücken, in denen spezifische Kundenanforderungen nicht erfüllt werden. Diese können die Geschäftslogik, die Integration der neuesten Algorithmen auf Grundlage künstlicher Intelligenz (KI), Maschinen- und Geräteschnittstellen usw. betreffen.

Daher müssen die Kunden zwangsläufig die Möglichkeit haben, ihren Code selbst weiterzuentwickeln, auch wenn dies über den genauen Umfang der app-basierten Lösung hinausgeht. Dazu gehört der Bedarf an Konnektivität mit anderen Lösungen und Datenquellen, das Füllen von Lücken bei spezifisch benötigten Funktionen sowie die Triage des Supports, wenn zentrale Lösungsbibliotheken aktualisiert werden. Es wird also Code benötigt, der entweder intern, von einem externen Anpassungspartner, einer Community oder durch die Verwendung von Middleware von Drittanbietern entwickelt wird. Solche Notlösungen sind selten ausgereift genug, um zukünftige Anforderungen oder flexible Anwendungsfälle abzudecken. **App-basierte Plattformen sind keine "no-code"-Lösung, sondern bringen uns dieselben Entwicklungs- und Supportprobleme, mit denen die frühen Softwareentwickler konfrontiert waren, wenn auch weitaus attraktiver.** Der benötigte Code wird vom Anbieter nicht zur Verfügung gestellt und fällt somit nicht in seinen Verantwortungsbereich.

Extrem: App-basierte Technologie

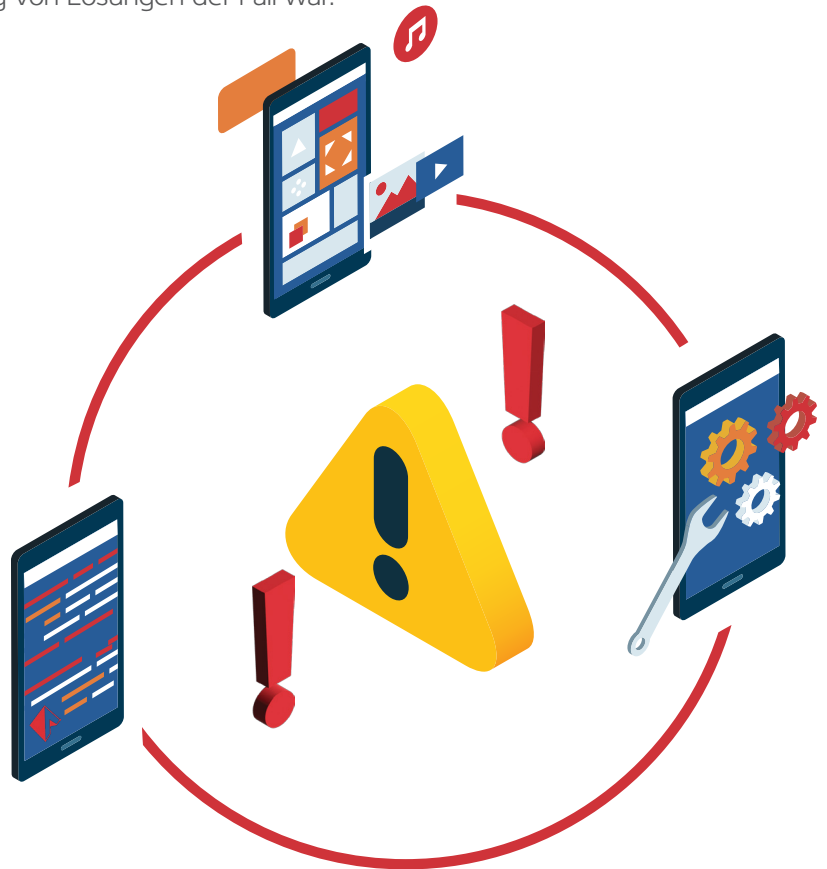
Dann wird argumentiert, dass die Kernfunktionalität der Bibliothek so weit wie möglich eingefroren werden sollte, wobei nur schrittweise, abwärtskompatible Änderungen vorgenommen werden sollten. Jedes Fertigungsunternehmen hat jedoch die Aufgabe, sich weiterzuentwickeln und zu expandieren, um neue Kunden und Geschäftsmöglichkeiten zu erschließen. Die steigende Anzahl der hergestellten Produkte, die Anzahl der Varianten, die Produkttypen in verschiedenen Sektoren (mit jeweils unterschiedlichen Betriebs- und Ausnahmeflussmodellen) sowie unterschiedliche Materialtypen, Automatisierungen, Flexibilitätsanforderungen, Häufigkeit von technischen Änderungen, Losgrößen, Qualitätsmanagementrichtlinien, Konformitäts- und Rückverfolgbarkeitsanforderungen sind nur einige der vielen unvermeidlichen Quellen für Abweichungen in den verschiedenen Fabriken. **Die Unfähigkeit - oder der extreme Aufwand -, app-basierte Plattformen entsprechend dieser sich ändernden Anforderungen anzupassen, wird daher zu einer erheblichen geschäftlichen Einschränkung.** Selbst die kleinsten Abweichungen können die Implementierung einer einfachen, aufgabenorientierten Lösung scheitern lassen. Einfache, app-basierte Modullösungen sind solchen Herausforderungen nicht gewachsen und orientieren sich daher erfolgreich nur an kurzfristigen, einfachen Produktionsszenarien.

Wenn man tiefer geht, verschlechtert sich die Situation zusehend. App-basierte MES-Lösungen erfordern in der Regel die Konfiguration von Datenelementen und -beziehungen durch den Kunden, wobei nur sehr wenig oder gar keine Kontrolle über die Benennung, Formatierung, Spezifikation von Beziehungen und Abhängigkeiten besteht und es besteht die Gefahr von Duplikaten, Überschneidungen und Inkonsistenzen zwischen verschiedenen Anwendungen. Die Verantwortung für die Ontologie und die Datenmodelle liegt letztlich beim Kunden, der in der Regel selbst weder Datenwissenschaftler noch Lösungsarchitekt ist. Vom Kunden erstellte Datenmodelle sind wahrscheinlich nicht so robust, dass sie den unmittelbaren Anforderungen genügen oder flexibel genug sind, um künftige Möglichkeiten zu unterstützen, wie dies bei der Eigenentwicklung von Lösungen der Fall war.



Die größten Herausforderungen bei extremer Lösungsdominanz

-  Veralteter app-spezifischer Code
-  Schwierigkeiten bei der Anpassung
-  Unvorhersehbares App-Verhalten
-  Bedarf an Code-„Pflastern“
-  Unzureichend definierte Datenmodelle
-  Sicherheitsschwachstellen



Das Fehlen eines robusten Datenmodells wirkt sich auch auf einen anderen Bereich aus – die Verbindung von App-basierten Lösungen mit anderen MES-Lösungen, die oft als ein starker Aspekt im Zusammenhang mit App-basierten MES-Implementierungen genannt wird. Eine erfolgreiche Integration setzt jedoch voraus, dass die Datenmodelle der verbundenen oder assoziierten Lösungen aufeinander abgestimmt sind, um den freien Informationsfluss genauso zu ermöglichen, wie es die Geschäftslogik der jeweiligen Anwendung erfordert. Dies führt zu einer sehr komplexen Integrationsarbeit, die eine umfangreiche Code-Entwicklung erfordert.

Mit einem solchen Durcheinander aus Code, der verschiedene konfigurierte Anwendungen, Open-Source-Funktionen, interne oder ausgelagerte Anpassungen und Middleware enthält, entsteht ein Minenfeld aus Sicherheitsproblemen und Schwachstellen. IT-Teams müssen in der Fertigung eine wichtige Rolle spielen und das "unsichere Terrain" der Digitalisierung des betrieblichen Netzwerks sichern. Da die Produktion Daten verbraucht und erzeugt, ist der Datenverkehr zwischen den OT- und IT-Netzwerken unausweichlich. Die Sicherheitsschwachstellen, die den Produktionsbetrieb so leicht unterbrechen, geistiges Eigentum nach außen dringen lassen und die Produktqualität beeinträchtigen können, müssen extrem gut kontrolliert und verwaltet werden, um erfolgreiche Cybersecurity-Angriffe zu vermeiden. Der App-basierte Plattformansatz geht in die entgegengesetzte Richtung, wobei einige sogar damit werben, dass die Lösung weiter von der IT-Sicherheit, Wartung und Kontrolle entfernt ist, was für viele ein Problem darstellt.

Das bedauerliche Ergebnis ist, dass die einfachere App-basierte Umgebung zwar anfänglich überzeugend sein mag, die längerfristigen Kosten und Risiken jedoch nicht tragbar sind, was von den Anbietern solcher Lösungen nicht vollständig offengelegt wird.

Der App-basierte Plattformansatz geht in die entgegengesetzte Richtung, wobei einige sogar damit werben, **dass die Lösung weiter von IT-Sicherheit, Wartung und Kontrolle entfernt ist, was für viele ein Problem darstellt.**



Extrem: Interoperabilität

Bei den anderen bisher betrachteten Extremen der Softwareentwicklungsstrategie tauchen unabhängig vom gewählten Ansatz immer wieder die gleichen Probleme auf. **Wird es jemals einen Weg geben, die Probleme zu überwinden, die der MES-Umgebung scheinbar anhaften?** Extreme Interoperabilität ist das Gegenteil von extremer Dominanz und bildet einen Rahmen, in dem die kontinuierliche Auswahl des "besten Werkzeugs für die jeweilige Aufgabe" erfolgen kann, entweder innerhalb einer einzigen flexiblen Gesamtlösung oder mit der Freiheit der Auswahl von Lösungen verschiedener Marktteilnehmer, die jeweils ihre eigene Nische von Funktionen unterstützen.

Der Wert dieses Ansatzes ist die Fähigkeit, Lösungen zu kombinieren, die genau den Anforderungen entsprechen, konfliktfrei zusammenarbeiten und jederzeit problemlos unabhängig voneinander aktualisiert werden können. Eine derartige Interoperabilität ist ein recht neues Konzept und gewinnt rasch an Bedeutung. Aus Sicht der digitalen Transformation wurde eine der größten Herausforderungen für die Interoperabilität (nämlich der Austausch von Fertigungsdaten) beispielsweise durch die Verwendung des IIoT-Nachrichten-basierten IPC Connected Factory Exchange (CFX)-Standard gelöst, bei dem ein einziges Datenmodell für alle Maschinen und Vorgänge verwendet wird, wodurch eine vollständige Plug-and-Play-Interoperabilität ermöglicht und herstellerbezogene Abhängigkeiten vermieden werden.

Ein gewisses Maß an Interoperabilität ist erforderlich, um aktuelle und künftige KI-basierte Anwendungen in bestehende MES-Rahmenwerke zu integrieren. **KI-Anwendungen übernehmen das "Denken", während das MES die Aktionen verwaltet.**

Die Lektion, die wir aus dem CFX-Beispiel gelernt haben, ist, dass die Interoperabilität auf offenen Standards beruhen muss. Andernfalls wird die Konnektivität durch proprietäre Schnittstellen und/oder Datenmodelldarstellungen eingeschränkt, was wiederum die Notwendigkeit von Anpassungen, Datenmodellübersetzungen und den Einsatz von Middleware nach sich zieht.

Die rasante Entwicklung von KI-Anwendungen im Zusammenhang mit Fertigungsprozessen ist im vollen Gange. Dazu gehören Anwendungen im Bereich des maschinellen Lernens (ML), Closed-Loop-Feedback-Lösungen und die Live-Analyse von Mustern und Trends in Daten zu Themen wie vorausschauende Wartung, Energieeinsparung, Null-Fehler, Engpassbeseitigung usw.

Ein gewisses Maß an Interoperabilität ist erforderlich, um aktuelle und zukünftige KI-basierte Anwendungen in bestehende MES-Frameworks zu integrieren. KI-Anwendungen übernehmen das "Denken", während das MES die Aktionen verwaltet. Interoperabilität ist in diesen Bereichen, die sich erst noch entwickeln müssen, unerlässlich. Proprietäre "Ökosysteme" bergen ein sehr hohes Risiko, wenn sie nicht auf offenen Standards beruhen. Solche Ökosysteme eignen sich zwar gut für die Erprobung von Konzepten, bringen aber Aspekte extremer Beherrschung mit sich, indem sie Abhängigkeiten von mehreren Integrationspunkten mit jeder Lösung schaffen.

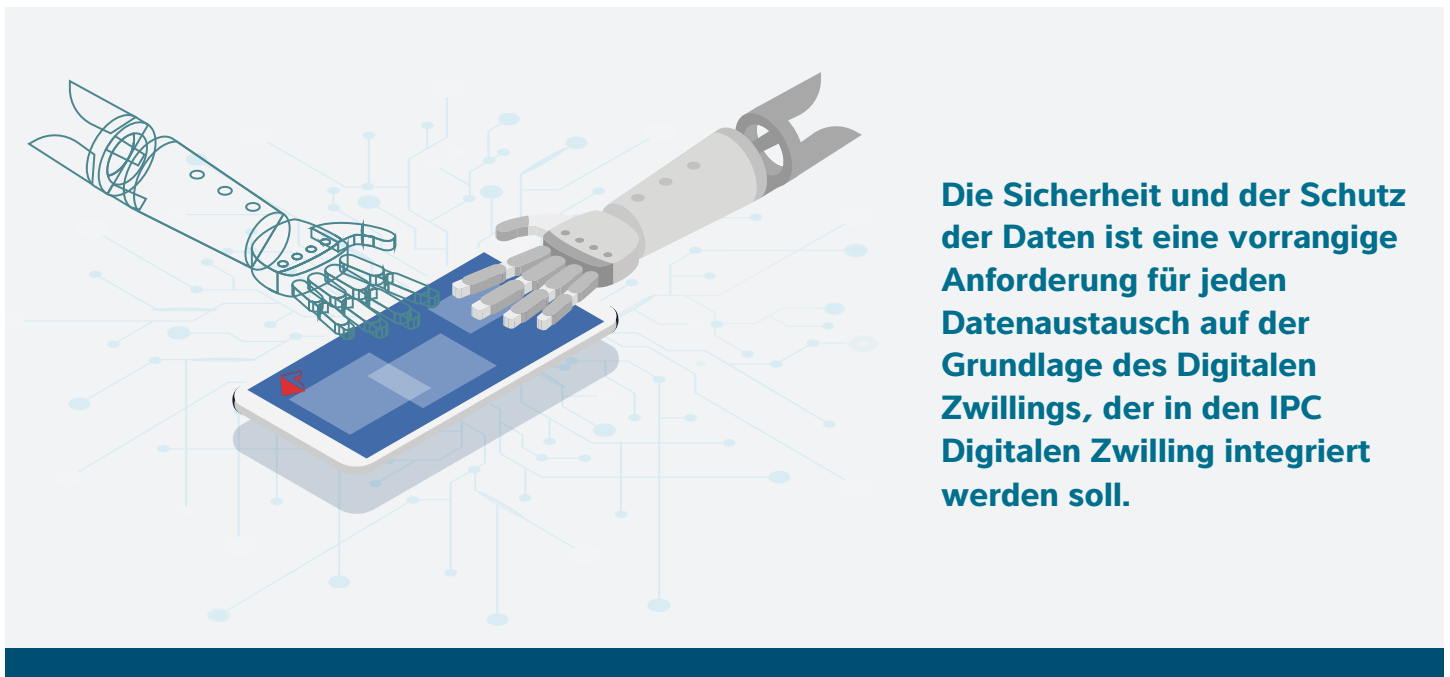


Ein gewisses Maß an Interoperabilität ist erforderlich, um aktuelle und künftige KI-basierte Anwendungen in bestehende MES-Rahmenwerke zu integrieren. KI-Anwendungen übernehmen das "Denken", während das MES die Aktionen verwaltet. Interoperabilität ist in diesen Bereichen, die sich erst noch entwickeln müssen, unerlässlich. Proprietäre "Ökosysteme" bergen ein sehr hohes Risiko, wenn sie nicht auf offenen Standards beruhen.

In oben genanntem Beispiel des Ansatzes zur extremen Dominanz wurden drei der fünf MES-Anforderungen des Kunden sehr gut von einem Anbieter unterstützt. Mit extremer Interoperabilität wäre es möglich, Standardlösungen von anderen Nischenanbietern zu integrieren, so dass alle fünf ursprünglichen Anforderungen exakt erfüllt und erfolgreich zusammen betrieben werden können, ohne dass interne Entwicklungen, kommerzielle Anpassungen oder der Einsatz von Middleware erforderlich sind. Jede Nischenlösung kann so ausgewählt werden, dass sie genau den Anforderungen in Bezug auf das erforderliche Funktionsniveau entspricht - wenn beispielsweise eine sehr hoch entwickelte MES-basierte Qualitätslösung benötigt wird, aber eine einfache visuelle Lösung für die Planung und Disposition ausreicht. **Jedes Element einer MES-Gesamtlösung kann nach dem genauen Bedarf spezifiziert werden, wodurch die Kosten und die Komplexität der digitalen Transformation reduziert und die Redundanz von zu komplexen oder sich überschneidenden Lösungen vermieden wird.**

Ein weiteres Beispiel für einen bestehenden offenen Standard, der eine extreme Interoperabilität unterstützen soll, ist der IPC Digital Twin, der den Inhalt des digitalen Zwillings während der Entwicklung, Herstellung und des Betriebs von Produkten auf dem Markt beschreibt und indiziert. Der digitale Zwillingsstandard des IPC fördert die Interoperabilität von Anwendungen in der gleichen Weise wie CFX die Interoperabilität für die Datenübertragung, wenn möglich in den bestehenden gängigen Datenformaten. Die Sicherheit und der Schutz der Daten ist eine vorrangige Anforderung für jeden auf dem Digitalen Zwilling basierenden Datenaustausch, der in den digitalen Zwilling des IPC integriert werden soll.

Viele Organisationen entwickeln derzeit ihre eigenen Methoden zur Darstellung von Daten über "Dinge" in einem digitalen Zwillingsformat, indem sie wichtige Baugruppen und Maschinen in ihre Basiskomponentenelemente zerlegen, zu denen Daten erstellt werden, die ihre Form und Funktion beschreiben. Die Ergebnisse dieser Projekte bieten eine Auswahl an Formaten, von denen jedes durch die Erstellung einer Zelle innerhalb der digitalen Zwillingsstruktur des IPC deklariert werden kann, ähnlich wie eine moderne Mediendatei Metadaten über das Format, die Kodierungsmethode, die Bitrate usw. von Videodaten enthält.



Die Sicherheit und der Schutz der Daten ist eine vorrangige Anforderung für jeden Datenaustausch auf der Grundlage des Digitalen Zwillings, der in den IPC Digitalen Zwilling integriert werden soll.

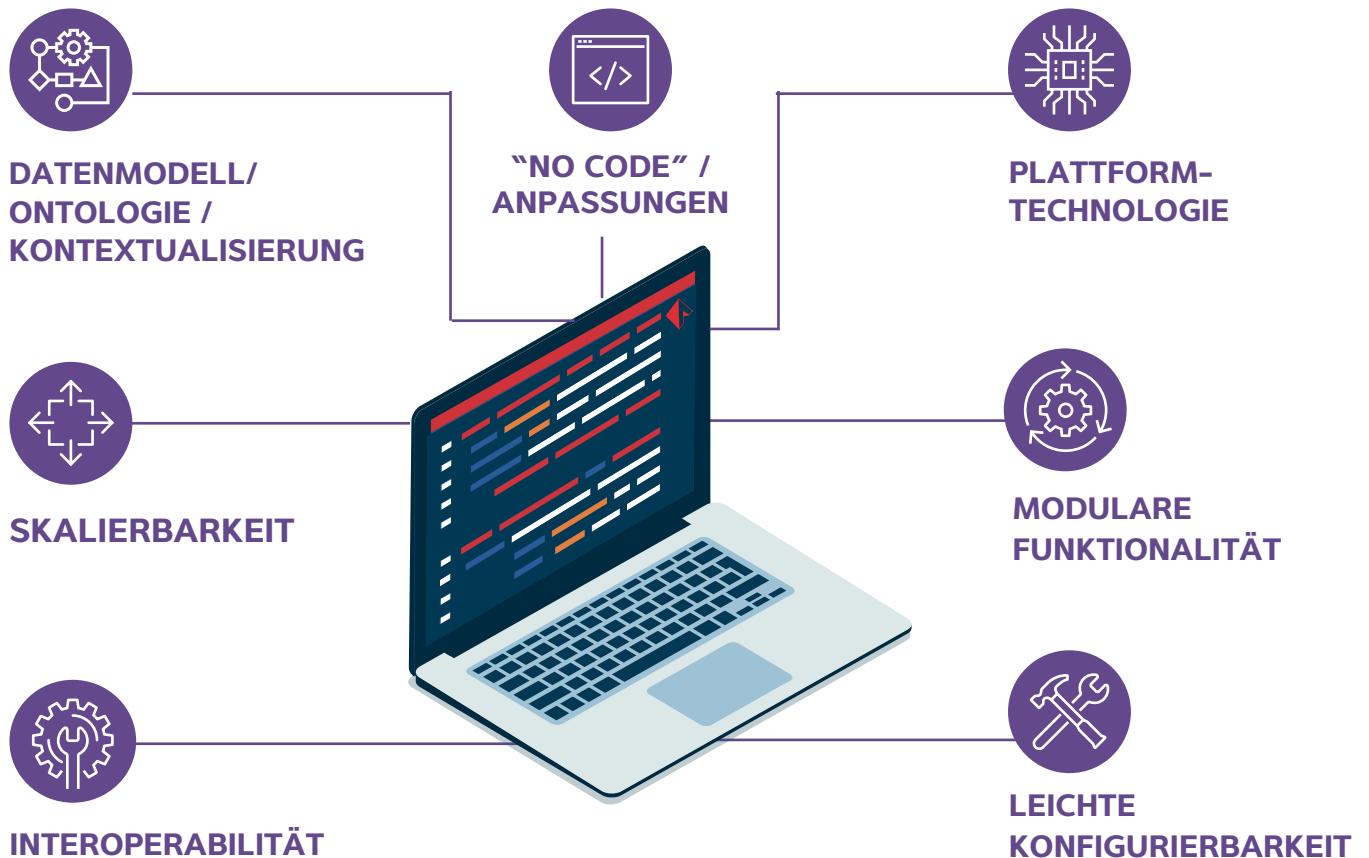
Extrem: Flexibilität

Wie wir gesehen haben, gibt es für jede extreme Herangehensweise an die Softwareentwicklung und an Lösungen zwingende Gründe für die Einführung, hinter denen sich sehr bedeutende, langfristig geschäftseinschränkende Nachteile verbergen, die in der Regel die Vorteile weit überwiegen. Jede MES-Lösung ist komplex. **Selbst diejenigen, die für die Implementierung an einem einzigen Standort entwickelt wurden, müssen die Flexibilität haben, sich im Laufe der Zeit zu entwickeln, wenn sich die Produktionsanforderungen ändern.** Aus der Perspektive einer kommerziellen Lösung sehen wir, dass keine zwei Fertigungsbetriebe - innerhalb desselben Unternehmens - gleich sind.

Die Vereinfachung von MES zu einer Low-Level-Lösung, die auf gemeinsamen Elementen basiert, wie im Fall der App-Plattform, ist nicht realistisch, da das Datenmodell, die Ontologie, die Verbindungen und die Integrationen eine einzigartige Anforderung für jeden Fertigungsbetrieb bleiben. Sie werden von der Lösung selbst nicht erfüllt, was bedeutet, dass Investitionen in die Code-Entwicklung, Middleware und ein erhebliches Lösungs-Engineering dem Kunden überlassen werden, die er erst nach dem Kauf entdeckt. Keines der Probleme mit dem Code wurde wirklich gelöst.

Wenn all dies zutrifft, wie können Sie die Kosten und Risiken von MES-Lösungen am besten minimieren? **Flexible MES-Lösungen enthalten viele Konfigurationsoptionen, mit denen die Arbeitsweise maßgeschneidert werden kann. Sie stellen eine Auswahlmöglichkeit für allgemeine betriebliche Funktionen dar und erfüllen auch erweiterte Anforderungen.** Diese Konfigurationsoptionen sind auf den ersten Blick und ohne Anleitung unübersichtlich, stehen aber für die Nachhaltigkeit der Lösung, da sich das MES an erhebliche Produktionsänderungen und die Einführung von Best Practices anpassen kann. Solche Optionen tragen dazu bei, die digitale Transformation voranzutreiben, indem sie die Möglichkeit bieten, im Laufe der Zeit bessere Verfahren zu übernehmen.

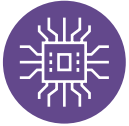
Für die MES-Flexibilität sind folgende Überlegungen besonders wichtig:





DATENMODELL / ONTOLOGIE / KONTEXTUALISIERUNG

Die wichtigsten Werte einer MES-Lösung lassen sich nicht einfach mit einer Liste und angekreuzten Kästchen ermitteln. **Die Intelligenz einer Lösung wird durch das Verständnis des Zusammenspiels der Tausenden von Datenpunkten bestimmt, die kontinuierlich aufgezeichnet werden, die Regeln und Algorithmen ermöglichen, die Informationen für die Sichtbarkeit, Verwaltung und Optimierung des Produktionsbetriebs in Echtzeit bereitstellen.** Der Beweis für eine solche Technologie ist, dass die Software Produktionsherausforderungen löst, bevor diese relevant werden, anstatt bestehende Abläufe ohne zusätzliche Unterstützung einfach zu automatisieren. Das Datenmodell ist der wichtigste Ausgangspunkt für die Intelligenz einer MES-Lösung.



PLATTFORMTECHNOLOGIE

Die zugrundeliegende Plattformtechnologie sollte **auf einer singulären, vorzugsweise IIoT-basierten Technologie aufbauen.** Dadurch werden Szenarien vermieden, bei denen ältere Lösungen so kombiniert wurden, dass Integration vorgegaukelt wird, die in Wahrheit Synchronisierung ist. Die zugrundeliegenden Unterschiede zwischen den Lösungen führen zu Inkonsistenzen im Betrieb.



MODULARE FUNKTIONALITÄT

In der Branche gibt es viele unterschiedliche Meinungen darüber, was der wahre Umfang einer MES-Lösung ist. MES-Anwender sollten sich nicht um diese Argumente kümmern, **sondern die Lösung auf ihre aktuellen Bedürfnisse und künftigen Erwartungen abstimmen.** Eine MES-Lösung muss Modularität bieten, die eine kontrollierte, schrittweise Einführung je nach Kundenbedarf ermöglicht.



SKALIERBARKEIT

MES-Lösungen **sollten sowohl in Bezug auf die Funktionstiefe als auch auf die Anzahl der Benutzer skalierbar sein.** Die Anforderungen in der Fertigung werden stark von der Art und Klassifizierung des Produkts bestimmt. Es ist ganz normal, dass z. B. für das Qualitätsmanagement eine sehr hohe Spezifikation erforderlich ist, während für die Terminplanung, die Sichtbarkeit und die Durchsetzung des Routings nur die wesentlichen Tools benötigt werden.



LEICHTE KONFIGURIERBARKEIT

Genauso wichtig wie der tägliche Umgang mit der Lösung ist **eine eindeutige und intuitive Konfigurierbarkeit.** Die überwiegende Mehrheit der Softwarelösungen, selbst solche Pakete wie Microsoft Excel, werden nicht in vollem Umfang genutzt, weil das Bewusstsein und das Verständnis für die verfügbaren Merkmale und Funktionen fehlt.



INTEROPERABILITÄT

In dem Maße, wie sich Standards zur Förderung der Interoperabilität durchsetzen (z. B. der IIoT-Messaging-Standard IPC CFX und die Interoperabilität von Daten zwischen Lösungen), verringert sich die Notwendigkeit der Entwicklung maßgeschneiderter Schnittstellen und wird schließlich ganz entfallen. Es sollte nach Lösungsanbietern gesucht werden, die eine **Vorreiterrolle bei solchen Standards einnehmen und gegebenenfalls auch native Unterstützung bieten.**



“NO CODE” / ANPASSUNGEN

In einer perfekten Welt sollte für die Einführung einer MES-Lösung keine Code-Entwicklung erforderlich sein. Da sich die Interoperabilität in der gesamten Branche weiterentwickelt, sind Anpassungen nach wie vor auch für app-basierte Plattformlösungen erforderlich, wenn auch außerhalb der Kernlösung. Anwendungsspezifischer Code, ob innerhalb oder außerhalb der Kernlösung, sollte nach Möglichkeit vermieden werden, da er ein hohes Ausfallrisiko und hohe Betriebskosten birgt, da er oft nicht ordnungsgemäß gewartet oder bei der Weiterentwicklung von Lösungen berücksichtigt wird. **Jeder Code, der als Zusatz erforderlich ist, sollte in die Kernlösung selbst integriert werden, damit er als Teil der ganzheitlichen Lösung gewartet wird.** Durch ein derartiges Vorgehen wird auch die doppelte Entwicklung von Code vermieden und der Wert der Lösung für alle Benutzer erhöht. Auf diese Weise sehen wir die Entwicklung von MES-Lösungen als Teil des gesamten digitalen Transformationsprozesses, im Gegensatz zur Stagnation der Funktionalität, bei der die Anpassung eine reine „Bastelarbeit“ für spezifische Kundenanwendungsfälle bleibt.

Fazit

Die Betrachtung der extremen Ansätze für MES-Software zeigt, dass bei der Auswahl von Lösungen einige wichtige Entscheidungen zu treffen sind, die weit über die Marketingbotschaften hinausgehen, die ebendiese Ansätze hervorheben. Ein einfacher, auf dem kleinsten gemeinsamen Nenner basierender Lösungsansatz für MES ist niemals realistisch. Jeder angebliche Ansatz mit der Behauptung "kein Code" verlagert den unvermeidlichen Kodierungsbedarf aus der Kernlösung heraus, was die Verwaltung erschwert, und zwar auf Kosten eines unklaren oder nicht robusten integrierten Datenmodells.

Die Kunden sollten nicht an eine begrenzte proprietäre Lösung eines marktbeherrschenden Anbieters gebunden sein und teuer für fehlende Funktionen bezahlen müssen, die für immer kostspieligen kundenspezifischen Support erfordern.

Solange sich eine echte, auf Standards basierende Interoperabilität nicht durchgesetzt hat, ist eine vollständige Interoperabilität unerreichbar - ein Ziel, das das auf offenen Standards basierende Ökosystem von MES- und KI-Anwendungen anstrebt.

MES-Lösungen sollten einfach "out of the box" funktionieren und auf einer einfachen Auswahl von High-Level-Optionen basieren, Werten, die auf einem ausgereiften Datenmodell basieren, und unterstützt durch eine integrierte Ontologie, die auf vielen Jahren Erfahrung in der Fertigung basiert, mit der Flexibilität, sich weiterzuentwickeln. Nur dann können die vielen individuellen Kundenbedürfnisse erfüllt werden, während gleichzeitig das Versprechen eines Mehrwerts im Einklang mit den Erwartungen der digitalen Transformation erweitert und weiterentwickelt wird.

Weitere Informationen



Email: infode@aiscorp.com

Webseite: www.aiscorp.com/de

@FactoryLogix

linkedin.com/company/aegis-industrial-software

Hauptsitz des Unternehmens

220 Gibraltar Road, Suite 300
Horsham, PA 19044

Telefon +1.215.773.3571

Firmensitz in Europa

Wetterkreuz 27
91058 Erlange

Telefon: +49.913 1.7778.10

Firmensitz in Asien

Rm. 809, Dahua Hucheng Business Center
No 6, Lane 239, Dahua No. 1 Road
Putuo District, Shanghai, 200442, P.R. China

Telefon: +86 2 1 5882 4882