

Continuous Engineering für das Internet of Things

Bewährte Methoden nutzen
für neue Herausforderungen
in der Entwicklung

Inhaltsverzeichnis

3	Einleitung
4	Die Besonderheiten und Herausforderungen von IoT-Produkten und -Services
5	Continuous Engineering
6	Systems Engineering für das IoT
7	Entwicklung von Embedded Software für das Internet der Dinge (IoT)
8	Entwicklung von IoT-Anwendungssoftware
9	Continuous Engineering und agile Methoden
9	Die IBM Internet of Things Continuous Engineering-Lösung
10	Fazit
11	Weitere Informationen

Übersicht

Das Internet der Dinge oder Internet of Things (IoT) hat den Sprung vom vielfach erwarteten Konzept zur spannenden Realität geschafft. In zunehmendem Maße wird einfach erwartet, dass Dinge vernetzt sind, und dadurch verändert sich unumkehrbar die Art und Weise, wie wir mit der physischen Welt interagieren. Infolgedessen können Unternehmen reibungsloser und profitabler arbeiten und sich sogar neu erfinden. Die vernetzten Dinge um uns herum können unser Leben und unsere Arbeit leichter, sicherer und komfortabler machen – und das IoT kann uns helfen, große Probleme anzugehen, beispielsweise bei Umweltverträglichkeit und bei der Behandlung chronischer Krankheiten.

Doch die zunehmende Komplexität vernetzter Systeme und ihre inhärente Neigung zu kontinuierlicher Veränderung führt auch zu neuen Arten von Fehlern, die zudem noch kostspieliger werden können. Diese neuen Rahmenbedingungen bedeuten neue Herausforderungen für Unternehmen in den Bereichen Engineering und Fertigung. Gleichzeitig ist es aufgrund der komplexen Entwicklungsumgebung und der sich schnell weiterentwickelnden Anforderungen der IoT-Ära erforderlich, dass Entwicklungsteams ihr Wissen ständig erweitern und sich schnell anpassen können.

Es gibt eine unglaubliche Vielfalt an IoT-Lösungen, von neuen Produkten, die erstmals vernetzt werden, bis hin zu Systemen, die für das Funktionieren der Infrastruktur entscheidend sind, wie beispielsweise Transport- oder Energiesysteme. Auch die Relevanz der Konnektivität kann enorm variieren. Konnektivität kann ein angenehmes Plus darstellen oder auch eine entscheidende oder gar lebensrettende Fähigkeit sein. Im Großen und Ganzen ist es so, dass das IoT eine immer bedeutendere Rolle auf der Welt spielt, weshalb auch die Zuverlässigkeit und Sicherheit wichtiger werden. Es gibt zwar keine Universal-Entwicklungslösung für das ganze Spektrum an IoT-Systemen, doch solide Engineering-Praktiken sind in praktisch jedem Fall unerlässlich. An dieser Stelle kann das Systems Engineering eine wichtige Rolle spielen.

Das Systems Engineering ist eine Fachrichtung, die ein besonderes Augenmerk auf die Koordinierung und Verwaltung der Komplexität und wechselseitige Abhängigkeiten von Systemen, Subsystemen und Komponenten legt, um sicherzustellen, dass die gewünschten Funktionen bereitgestellt werden und das System zuverlässig ist. Das Systems Engineering und das Denken in Systemen waren lange eine Kernkompetenz der Luft- und Raumfahrtindustrie. Doch auch in der sich schnell verändernden Welt der IoT-Lösungen können sich daraus große Vorteile ergeben.

Die zentralen Eckpunkte eines Systemansatzes bestehen in der Koordinierung und Integration voneinander abhängiger Subsysteme. Dieses Prinzip kann auf die Entwicklung von Embedded Software und Anwendungssoftware für IoT-Systeme ausgeweitet werden. IoT-Konnektivität wird zunehmend zu einem Hauptunterscheidungsmerkmal für neue Produkte. Die Software, die diese Konnektivität ermöglicht, zeichnet sich durch verschiedene Umgebungen, ein hohes Maß an Komplexität und schnelle Veränderungen aus, was die Entwicklung zu einer großen Herausforderung macht. Um diese Herausforderung zu bewältigen, kann der Systemansatz ergänzt werden durch agile Methoden und Continuous Engineering, die in der Kombination zu einer besseren Reaktionsfähigkeit, einer besseren Qualitätskontrolle, höherer Produktivität und kürzeren Lieferzeiten führen können.

Für eine effektive Entwicklung von IoT-Lösungen sind Prozesse erforderlich, die sowohl robust als auch flexibel und anpassbar sind. Das Continuous Engineering und agile Prinzipien liefern bereits die wesentlichen Grundlagen dafür und können mit agilen Frameworks wie SAFe und Engineering-Lösungen von IBM® weiter optimiert werden. Aus allen diesen Gründen ist der Systems-Engineering-Ansatz gut für den Einsatz in softwareintensiven IoT-Systementwicklungsumgebungen geeignet.

Einleitung

Der Hype rund um das IoT weicht nun schnell der Realität implementierter Produkte und Services.

Den Analysten von IDC zufolge sollen die Ausgaben auf dem IoT-Markt von rund 690 Mrd. USD im Jahr 2015 auf rund 1,46 Bio. USD im Jahr 2020 ansteigen, mit einer jährlichen Wachstumsrate von 16,1 %. Die Anzahl an installierten IoT-Endpunkten wird von 12,1 Mrd. im Jahr 2015 auf über 30 Mrd. im Jahr 2020 ansteigen.¹

Die Konnektivität hat den Status, lediglich ein interessantes Feature zu sein, hinter sich gelassen und ist heute in vielen Märkten die Grundvoraussetzung für ein Produkt, das konkurrenzfähig und differenzierbar ist.

IoT-Produkte und -Services können ebenso elementar wie kritisch sein: kostenkritisch, verfügbarkeitskritisch, markenkritisch und sogar sicherheitskritisch. Deshalb müssen die Hersteller von Produkten und Anbieter von Services die Herausforderungen des Engineerings für das IoT verstehen und entsprechend darauf reagieren. Die Konnektivität ermöglicht nicht nur die Erweiterung der Fähigkeiten und Funktionen von IoT-Produkten und -Services, durch sie erhöht sich auch deren Komplexität. Neue Funktionen bringen neue Arten von Fehlern mit sich. Zusätzliche Komplexität kann ohne angemessenes Management die Wahrscheinlichkeit des Auftretens von Fehlern steigern. Zudem können die Folgen von Fehlern auch nur schwer vorhersagbar sein.²

Daher erfordern zunehmend wichtigere Produkte und Services ein robustes IoT-Engineering. Die wesentlichen Herausforderungen umfassen Folgendes:

- Bereitstellung einer überzeugenden Funktionalität (wobei sich die Anforderungen kontinuierlich ändern können)
- Bereitstellung angemessener Verlässlichkeit in Form von Sicherheit (keine Schädigungen/Unversehrtheit), Zuverlässigkeit (Verfügbarkeit von Services) und Schutz (kein Eindringen, keine Beeinträchtigungen, kein Diebstahl)
- Bereitstellung der Lösung in einem offenen Kontext, in dem sich einige der mit der Lösung verbundenen Technologien und Komponenten nicht unter direkter kommerzieller oder technischer Kontrolle befinden
- Bereitstellung der Lösung mit einer angemessenen Geschwindigkeit und zu angemessenen Kosten, um Wettbewerbsbedrohungen und sich ändernden Marktanforderungen zu begegnen

Mit dem IoT verbundene Produkte und Anwendungen erfordern einen systemorientierteren Ansatz für das Engineering. Systemdenken, insbesondere das Konzept des sich ergebenden Verhaltens (sowohl gewollt als auch ungewollt), ist für eine hohe Qualität bei IoT-Entwicklung und -Design entscheidend. Das Systems Engineering – und insbesondere das System of Systems Engineering – kann dazu beitragen, die Agilität und Qualität von IoT-Entwicklung und -Design zu stärken, vor allem, wenn das konzipierte Produkt anderen Produkten entsprechen muss, die nicht vom Entwickler kontrolliert werden.

Nichtsdestotrotz müssen Systems-Engineering-Ansätze richtig dimensioniert sein, um für das IoT angewendet werden zu können. Sie müssen sich zwischen zwei Extremen bewegen: höchst agile Ad-hoc-Entwicklungsprojekte auf der einen Seite und akribisches und kostenintensives Systems Engineering auf Luftfahrtniveau auf der anderen Seite. Ein besonderes Augenmerk muss bei IoT-Systemen auf die Aspekte Sicherheit und Schutz gelegt werden, und zwar noch mehr als bei herkömmlichen Apps und Softwareprodukten. Die Tools, die solche Engineering-Ansätze unterstützen, müssen flexibel und integriert sein, sodass sie das richtige Maß an Kontrolle und Präzision ermöglichen und gleichzeitig die Erfordernisse kurzer Entwicklungszyklen und einer schnellen Markteinführung erfüllen.

Um die bestmöglichen Auswirkungen auf die IoT-Entwicklung zu erzielen, sollten Systems-Engineering-Ansätze Teil einer umfassenden Continuous-Engineering-Methodologie sein. Beim Continuous Engineering wird das Feedback von vernetzten Produkten und Systemen genutzt, um kontinuierlich Informationen für Produktverbesserungen und neue Designs zur Verfügung zu stellen. Es umfasst bewährte Prinzipien und Praktiken und kombiniert Systemdenken, Systems Engineering, Embedded-Software-Entwicklung und IoT-Anwendungssoftware-Entwicklung mit angemessener Automatisierung für die effiziente Umsetzung dieser Praktiken in realen Produktentwicklungsumgebungen. In diesem White Paper werden die Herausforderungen beschrieben, mit denen die Teams, die IoT-Lösungen entwickeln, konfrontiert sind. Außerdem wird erläutert, wie diese Herausforderungen mit einem agilen Continuous-Engineering-Ansatz bewältigt werden können.

Die Besonderheiten und Herausforderungen von IoT-Produkten und -Services

Ein Großteil der Wirkmacht des IoT ist auf die direkte Verbindung zurückzuführen, die es zwischen Kunden und Unternehmen ermöglicht. Diese Verbindung kann Geschäftsmodelle verändern, indem ganz neue Geschäftsprozesse ermöglicht oder Prozesse bedeutend effizienter als herkömmliche, nicht digitale Geschäftsprozesse gemacht werden. Es steht aber mehr auf dem Spiel, da der Kunde unmittelbar der Effektivität dieser digitalen Prozesse ausgesetzt ist.

Neben Kritikalitätscharakteristika gibt es bei IoT-Anwendungen in bestimmten Branchen verschiedene Dynamiken, die vorschreiben, wie schnell die Entwicklungsreaktion sein muss. Kleine Verbraucherprodukte und -geräte wie Smartphones oder Wearables können beispielsweise aufgrund des Marktwettbewerbs und sozialer Trends sehr schnelle Entwicklungszyklen erfordern. Im Gegensatz hierzu unterliegen die vernetzten Komponenten einer intelligenten Infrastruktur wie ein öffentliches Transportsystem bedeutend strategischeren Erfassungsprozessen und erfordern deshalb eine höchst robuste, aber weniger schnelle Entwicklung.

Weil das IoT ein relativ neuer und sich schnell verändernder Bereich der Technologie ist, sind Wesen und Charakter der IoT-Entwicklung nicht in ihrer finalen Ausgestaltung erkennbar. Es gibt aber verschiedene Arten von IoT-Herausforderungen:

- Einige IoT-Produkte und -Services sind native IoT-Anwendungen. Diese können ihre Funktionen nur dank IoT bereitstellen werden und Unternehmen, die derartige Produkte und Services anbieten, sind im Hinblick auf den Erfolg ihrer Marke in entscheidendem Maße vom IoT abhängig.
- Bei manchen Produkt- und Servicekonzepten werden sich die Funktionen und der Umfang ändern, sodass sie im Laufe der Zeit an Bedeutung gewinnen werden. Dementsprechend müssen sich dann auch die Entwicklungsprozesse weiterentwickeln und robuster werden.
- Bestimmungen und Standards können sich ändern, während Produkt- und Servicekonzepte sich stärker etablieren, und so zu neuen Anforderungen an Engineering und Entwicklung führen.
- Bei der Entwicklung eines vernetzten Produkts sind viele Parteien involviert – oftmals ein ganzes Netzwerk an Anbietern von Software- und Hardwaretechnologiekomponenten und Datenlieferanten. Zwischen diesen Parteien kann es wechselseitige Abhängigkeiten für den Erfolg geben; dennoch arbeiten sie nicht in demselben Rhythmus. Diese zusätzlichen Abhängigkeiten erfordern flexible Entwicklungsprozesse, um Änderungen zeitnah zu beurteilen und effektiv darauf zu reagieren.
- Die „Dinge“ im IoT müssen mit anderen vernetzten „Dingen“ in einem Ecosystem interagieren. M2M (Maschine-zu-Maschine)-Kommunikation und -Transaktionen werden im Laufe der Zeit geläufiger werden und an Bedeutung gewinnen.
- Die öffentliche Wahrnehmung ändert sich ständig: Absoluter Optimismus bezüglich IoT und kognitiver Systeme kann zu Skepsis werden bezüglich der Sicherheit und des Schutzes solcher Systeme und zu Besorgnis, ob es klug ist, die Einführung schnell voranzutreiben.

Die starke Verbindung mit dem Kunden, die durch das IoT ermöglicht wird, fördert nicht nur die Kundenbindung, sondern verändert auch die Produktentwicklung und -bereitstellung durch:

- *Schnelles Feedback*
Die Möglichkeit, Echtzeiterkenntnisse über die Produkt- und Serviceleistung und das Kundenverhalten zu gewinnen. Dieses Feedback gibt Unternehmen die nötigen Einblicke, um sich durch kontinuierliche Anpassung an Kundenanforderungen und Geschäftschancen von ihren Konkurrenten abzuheben
- *Schnelle Bereitstellung*
Die Möglichkeit, Kunden regelmäßig neue und überarbeitete Funktionen bereitzustellen und Wettbewerbsvorteile kontinuierlich zu optimieren.

Die vielen neuen Anwendungsfälle und Variablen, die IoT in die Produkt- und Serviceentwicklung einführt, erschweren die genaue Vorhersage, welche Fähigkeiten in der Entwicklung wann erforderlich werden. Eine robuste IT-Entwicklungsfunktion muss also zur Weiterentwicklung imstande sein, wenn die Entwicklungsanforderungen sich ändern.

Continuous Engineering

Technische Systeme mit Verbindung zur realen Welt bergen – anders als reine Softwareanwendungen – erhöhte Risiken für Menschen, Vermögenswerte und die Umwelt, wenn es zu einer Fehlfunktion kommt. Dieses Risiko wird bei IoT-Systemen noch verstärkt, da zumindest potenziell jedes IoT-Produkt jedes andere vernetzte Produkt auf der Welt beeinflussen kann! Ein zu rigoroser Ansatz in einem Kontext, in dem schnelle Produktupdates erforderlich sind, Produktfehlfunktionen aber nur geringe Kosten verursachen, könnte die Fertigstellung von Produkten unnötig verzögern, zur Aufgabe eines Wettbewerbsvorteils führen oder einen kommerziellen Fehlschlag zur Folge haben. Andererseits kann ein Mangel an Rigorosität in einer sicherheitskritischen oder einer anderen Umgebung mit hohen Kosten bei Fehlern noch größere Schäden nach sich ziehen. Und ein sich schnell änderndes Produktkonzept – bei dem die Marktgröße, die Kosten von Fehlern und das künftige Regulierungssystem Unbekannte sind – erfordert einen Ansatz, der sich auf effiziente Weise an die Anforderungen anpassen lässt, wenn diese sich ändern.

Systemdenken ist ein unverzichtbarer Aspekt der Entwicklung, der für eine erfolgreiche IoT-Entwicklung erforderlich ist: Man muss sich der Anforderungen für die Erstellung eines verteilten Systems von Systemen (System of Systems) bewusst sein und die Strategien in Engineering und Entwicklung darauf ausrichten. Je nach Kritikalität des Produkts oder Services kann dies von einer effektiveren Zusammenarbeit zwischen den verschiedenen involvierten Engineering-Teams und Organisationen bis hin zu bedeutend formelleren und rigoroseren Systems-Engineering-Methoden reichen.

Zudem ist die Softwareentwicklung für die IoT-Implementierung entscheidend, ganz gleich, ob es sich um die Entwicklung von Embedded Software für die Systemendpunkte („Dinge“) oder um die Entwicklung der übergreifenden IoT-Anwendungssoftware handelt.

Alle diese Engineering-Aktivitäten erfordern das richtige Maß an Zusammenarbeit, Nachverfolgbarkeit, Automatisierung, Reaktionsfähigkeit bei Veränderungen und Rigorosität je nach Kritikalität des Produkts oder Services, das bzw. der entwickelt wird.

Checkliste: Wie kritisch ist Ihre IoT-Lösung?

- Führt Ihr IoT-Produkt zu einer Veränderung Ihres Unternehmens, von einem Hersteller herkömmlicher mechanischer oder elektrischer Produkte zu einem Hersteller softwareintensiver Produkte? Verfügen Sie intern über die Fähigkeiten und das Wissen, um die Transformation zu bewerkstelligen?
- Kann Ihr vernetztes Produkt oder Ihr vernetzter Service direkte Schäden für Menschen oder Vermögenswerte verursachen? Beispiele hierfür wären ein vernetztes privates Sicherheitssystem oder Heizsystem oder eine autonome Fahrzeugsteuerung.
- Kann Ihr vernetztes Produkt oder Ihr vernetzter Service indirekte Schäden verursachen? Ein Beispiel hierfür wäre ein vernetztes Elektrizitätswerk, in dem eine Fehlfunktion zu Schäden an der Ausrüstung, zu Stromausfällen in Häusern und zur Beeinträchtigung der Sicherheit in der Bevölkerung führen könnte.
- Hängt die Benutzererfahrung komplett vom Aspekt der Vernetzung Ihres Produkts oder Services ab? Ein vernetzter Taxirufservice kann beispielsweise ohne Konnektivität nicht funktionieren. Ein vernetztes Fahrzeug hingegen funktioniert als Kfz jedoch auch dann, wenn keine Konnektivität gegeben ist.
- Wird Ihr IoT-Produkt oder -Service von Hardware- und Softwarekomponenten abhängig sein, die im Hinblick auf Engineering, Vermarktung oder beides nicht Ihrer direkten Kontrolle unterliegen?
- Ist dessen Verfügbarkeit von entscheidender Bedeutung? Ein intelligentes Zählersystem in einem Elektrizitätswerk würde beispielsweise nur mäßige Beeinträchtigungen für Benutzer verursachen, wenn der Energieverbrauch nicht richtig überwacht wird. Ein solcher Fehler wäre aber für das Netz katastrophal, wenn Bedarfsinformationen dem Versorgungsunternehmen nicht zur Verfügung stünden.
- Gibt es eine sehr große Anzahl an Benutzern oder wird sie wahrscheinlich sehr groß werden?
- Besteht finanzielle Kritikalität? Dies ist oftmals mit der Anzahl an Benutzern, der Sicherheitskritikalität oder Verfügbarkeitsanforderungen des Produkts oder Services oder dem Ausmaß der finanziellen Schäden im Falle eines Fehlers verbunden.

Continuous Engineering nutzt bewährte Praktiken für Systemdenken, Systems Engineering, Embedded-Software-Entwicklung und IoT-Anwendungssoftware-Entwicklung zusammen mit angemessener Automatisierung für die effiziente Umsetzung dieser Praktiken in realen Produktentwicklungsumgebungen.

Continuous Engineering ist kein „Alles oder nichts“-Ansatz. Es ist auf Skalierbarkeit gemäß den Anforderungen der Entwicklung von IoT-Produkten und -Services ausgelegt und kann in verschiedenen Dimensionen angepasst werden:

- Über verschiedene Teamgrößen und Entwicklungsnetzwerkstrukturen hinweg, vom Einfachen bis hin zum Komplexen
- Über verschiedene Entwicklungsbereiche des Systems Engineering, der Embedded-Software-Entwicklung und der IoT-Anwendungsentwicklung hinweg
- Über verschiedene Engineering-Aktivitäten hinweg, von der Anforderungsermittlung und dem Management bis hin zur Systemintegration und zu Tests

Durch Priorisierung der Anforderungen eines Projekts kann das Continuous Engineering dort angewendet werden, wo es die höchsten Nutzen bringt, und die Anwendung kann angepasst werden, wenn sich die Entwicklungsanforderungen und Prioritäten ändern.

Systems Engineering für das IoT

Entdecken Sie die Vorteile, die das Continuous Engineering im Hinblick auf das Systems Engineering für das IoT ermöglichen kann. [Klicken Sie hier, um sich ein Video anzusehen.](#)

Das Systems Engineering ist ein wesentlicher Aspekt einer übergreifenden Continuous-Engineering-Methodologie. IoT-Systems-Engineering befasst sich hauptsächlich mit der Koordinierung der Komplexität und Abhängigkeiten von IoT-Systemen, um sicherzustellen, dass die gewünschten Funktionen bereitgestellt werden und das System zuverlässig ist. Systemtechniker sind für die Erreichung von zwei Hauptzielen verantwortlich:

- Alle Systemanforderungen werden erfüllt, einschließlich Benutzer-, Markt- und regulatorische Anforderungen.
- Das System wird zeitgerecht und auf kosteneffiziente Weise bereitgestellt, sodass die technischen und kommerziellen Ziele des Unternehmens erreicht werden.

Systems Engineering muss ebenfalls sicherstellen, dass das System diese Ziele auch in einem dynamischen Bereitstellungskontext weiterhin erreicht. Das Systems Engineering findet bereits seit den ersten Phasen des amerikanischen Raumfahrtprogramms Anwendung und hat maßgeblich zu dessen Erfolg beigetragen. Seither hat es sich zu einer Disziplin entwickelt, die in vielen Branchen aktiv zum Einsatz kommt. Das Systems Engineering für das IoT muss äußerst agil sein, jedoch ohne die Haupteigenschaften zu vernachlässigen, die es so wertvoll machen.

Systemingenieure wissen, dass der Erfolg des Engineerings nicht nur von der ordnungsgemäßen Funktion eines verbundenen Systems abhängt, sondern auch von der Erfüllung wichtiger qualitativer Ziele wie Zuverlässigkeit, Elastizität, Sicherheit und Wartbarkeit. Jeder dieser sogenannten nicht-funktionalen Anforderungsbereiche erfordert eigene Entwicklungsverfahren, Werkzeuge und Verifizierungsansätze. Darüber hinaus müssen die Verifizierung und Validierung an die Herausforderungen der IoT-Systeme angepasst werden. Die Implementierung eines Systems, beispielsweise ein intelligenter Thermostat, mag einfach erscheinen, da der Thermostat nur für die Steuerung von Heizung und Kühlung des Gebäudes, in dem er installiert ist, verantwortlich ist. Wird der Thermostat jedoch über IoT vernetzt, ist ein umfassender Systems-Engineering-Ansatz erforderlich, um sicherzustellen, dass die Sicherheit von Menschen nicht gefährdet ist und die Gebäude, in denen das System installiert ist, keinen Gefahren durch Hacker, Kommunikationsprobleme oder fehlerhafte Softwareupdates ausgesetzt sind.

Systems Engineering hält viele Lösungen für die technischen Herausforderungen des IoT bereit, solange Systems Engineering selbst sich weiterentwickeln kann. Frühere Systems Engineering-Prozesse, die für langfristige, sorgfältig durchdachte Luft- und Raumfahrtprogramme gedacht waren, sind kein geeigneter Ansatz für dynamische IoT-Systeme. Entwicklungsteams können über verschiedene Standorte, Zeitzonen oder Länder verteilt sein. Märkte können gnadenlos sein, wenn Entwicklungen zu lange dauern. Und Kunden werden Schwierigkeiten und Probleme bei der Installation und dem Betrieb nicht lange tolerieren. Der Umfang an Systems Engineering-Praktiken, die für eine IoT-Anwendung zum Einsatz kommen, muss in sinnvoller Relation stehen zu den Risiken dieser IoT-Anwendung. Systemingenieure müssen agile, schlanke Techniken einsetzen, welche die erforderliche Strenge einbringen – jedoch ohne den Overhead wie in der Luft- und Raumfahrtindustrie.

Bestimmte Funktionen gelten für Systems Engineering allgemein, wie auch für die Entwicklung von Embedded Software und die Entwicklung von IoT-Anwendungen im Besonderen:

- *Zusammenarbeit, Workflow-Planung und Management*
Systemingenieure und anderen Ingenieursdisziplinen und Beteiligte haben zu jeder Zeit während des ganzen Projekts denselben aktuellen Stand. Bei Änderungen arbeitet das Team effektiv zusammen, ohne Doppelarbeit und ohne dass erforderliche Aufgaben übersehen werden.
- *Anforderungsmanagement und Rückverfolgbarkeit*
Bedarf und Anforderungen der Beteiligten werden korrekt erfasst, verstanden, aufgezeichnet und genutzt, um die Entwicklung geeigneter Lösungen voranzubringen. Rückverfolgbarkeit muss nicht nur zwischen Systems Engineering und Design, sondern auch zwischen nachgeschalteten Tests und der Integration gewährleistet sein.
- *Änderungs- und Konfigurationsmanagement*
Unterstützung verschiedener Produktversionen und -konfigurationen, welche die Mitglieder einer Produktfamilie unterscheiden
- *Modellbasiertes Systems Engineering*
Potenzielle Lösungen werden früh verstanden und evaluiert, und „was wäre, wenn ...“-Szenarien im Modell durchgespielt, um eine geeignete Lösung auszuwählen
- *Qualitätsmanagement des Systems*
Verifizierungs- und Validierungsaktivitäten werden von den Systemanforderungen abgeleitet und können bei Änderungen frühzeitig und häufig durchgeführt werden
- *Lebenszyklus-Informationsmanagement*
Durch detaillierte Rückverfolgbarkeit technischer Artefakte können die Entwicklungsergebnisse dank einer verbesserten Entscheidungsfindung optimiert werden
- *Agilität in Planung und Management*
Unterstützung bei der effizienten Aufspaltung der Entwicklung in einzelne Teilaufgaben, bei der dynamischen Priorisierung dieser Aufgaben und bei der iterativen Entwicklung, um getestete und funktionierende Komponenten zu liefern
- *Release- und Inbetriebnahmemanagement*
Die Balance finden zwischen schneller Lieferung von Updates mit neuen Funktionen der IoT-Infrastruktur und der Sicherstellung von Stabilität und Kontrolle bei der Inbetriebnahme
- *Kontinuierliche Rückmeldung*
Nutzen von Informationsquellen, wie operativen Leistungs-, Fehler- und Defektinformationen sowie Benutzerverhalten auf vernetzten Produkten und Systemen, zur kontinuierlichen Unterstützung der Produktverfeinerung und Designentwicklung.

Entwicklung von Embedded Software für das Internet der Dinge (IoT)

Entdecken Sie die Vorteile, die ein Continuous Engineering für die Entwicklung von Embedded Software für das IoT bringen kann.

[Klicken Sie hier, um sich ein Video anzusehen.](#)

Die Entwicklung von Embedded Software ist das Erstellen von Software für physische Geräte, welche die Komponenten eines IoT-Systems bilden. Die Geräte können von kleinen Sensoren bis zu komplexen physischen Produkten wie Flugzeugen und medizinischer Ausrüstung reichen. Embedded Software ist zunehmend für den Wert und die Differenzierung solcher Produkte verantwortlich. Es macht keinen Unterschied, ob diese Produkte IoT-Endpunkte sind. Jedoch verstärkt der IoT-Kontext viele der Herausforderungen für Entwickler von Embedded Software.

Bei IoT-Komponenten gibt es in der Regel mehr Abhängigkeiten als sie in anderen Bereichen der Softwareentwicklung anzutreffen sind:

- Die Betriebsumgebung von Embedded Software ist eng gekoppelt an die Hardware des Gerätes, d. h. sie ist vom Hardwaredesign abhängig
- Die Benutzererfahrung hängt entscheidend von der Embedded Software ab
- Eingebettete Funktionen werden durch Offboard-Funktionen der IoT-Anwendungssoftware ergänzt.

In einem IoT-Kontext gewinnt die embedded Entwicklung an Dynamik:

- Das Aktualisieren von Embedded Software ist keine Fertigungs- oder Serviceaktivität mehr sondern eine In-Service- oder sogar Echtzeit-Aktivität
- Betriebs- und Benutzerdaten können beinahe in Echtzeit wieder in die Softwareentwicklung eingespeist werden, um Produktverbesserungen anzustoßen. Dies erfordert einen Entwicklungsansatz, der schnell und effizient auf neue Informationen reagieren kann.
- Funktionen können auf neuen Wegen zwischen Embedded Software und Software, die auf einem Edge-Gerät oder Cloud-Server ausgeführt wird, verteilt werden.

Das Management der Abhängigkeiten, die Bewältigung der Dynamik der Entwicklung und die Unterstützung agiler Ansätze erfordern ein Zusammenspiel von Continuous-Engineering-Funktionen entsprechend dem Bedarf des spezifischen Projekts.

In regulierten (beispielsweise sicherheitskritischen) Umgebungen muss innerhalb der Entwicklungsprozesse nicht nur die Rückverfolgbarkeit gewährleistet sein, sondern es muss anhand dieser Rückverfolgbarkeit auch möglich sein, Entwicklungsartefakte zu extrahieren, um die Einhaltung von Standards nachzuweisen. Die Automatisierung sowohl bei der Erzeugung der Links für die Rückverfolgung wie auch beim Verfolgen dieser Links zum Extrahieren von Compliance-Berichtsinformationen ist von entscheidender Bedeutung für die Minimierung von Fehlern und die Eindämmung der Compliance-Kosten.

Entwicklung von IoT-Anwendungssoftware

Entdecken Sie die Vorteile, die Continuous Engineering bei der Entwicklung von IoT-Anwendungssoftware bringen kann. [Klicken Sie hier, um sich ein Video anzusehen.](#)

IoT-Anwendungssoftwareentwicklung steht für die Programmierung von Software zur Koordinierung der Komponenten eines IoT-Systems, um die Gesamtfunktion des Systems zu gewährleisten. Es handelt sich in der Regel um eine Kombination aus lokalen, cloudbasierten und mobilen Komponenten, die zahlreiche Technologien integriert, wie Datenerfassung und -speicherung, Analysen, Benutzer- und Zugriffsmanagement sowie die damit verbundenen Sicherheitsfunktionen und die Automatisierung von Geschäftsprozessen und weitere. IoT-Anwendungen gehen über herkömmliche Geschäftsanwendungen und Produktsoftware hinaus und sind zentraler Bestandteil eines Geschäftsmodells für ein IoT-basiertes Produkt bzw. einen Service. Daher sind Anwendungsstabilität und die Fähigkeit des Entwicklungsteams, auf sich ändernde Anforderungen zu reagieren, entscheidend für den Erfolg.

Auch wenn eine IoT-Lösung ein komplexes „system of systems“ darstellt, wird weiterhin erwartet, dass es in Echtzeit reagiert. Die IoT-Softwareentwicklung unterscheidet sich von der reinen Softwareentwicklung durch verschiedene Notwendigkeiten, unter anderem:

- Frühe Entwicklung der Architektur
- Fokus auf kritische Architekturmechanismen
- Berücksichtigung der physischen und virtuellen Inbetriebnahme
- Funktionale Integration mit Embedded Software in IoT-Endpunkte (oder „Dinge“)
- Berücksichtigung von Entwicklungsanalysen während des laufenden Entwicklungsprozesses.

IoT-Teams sind in der Regel voneinander abhängig und arbeiten verteilt und organisationsübergreifend. Sie können für längere Zeiträume Bestand haben.

Ein IoT-Entwicklungsprojekt ist mehr als die Summe vieler kleiner Softwareprojekte. Die Permutationen der Integrationen zwischen den Elementen des bereitgestellten Systems verursachen ein geometrisches Ansteigen der Komplexität. Die Anzahl der Variablen, wie Geräteplattformen, Microservices, Sprachen, Bibliotheken und vieles mehr, fügt dem ohnehin schon komplexen Prozess der Softwareentwicklung weitere Multiplikatoren hinzu. Im Ergebnis erfordert die Navigation über Artefakte, Werkzeuge und Zeitpläne hinweg mehr Intelligenz und Support.

Sicherheit, Datenschutz, Skalierbarkeit, Inbetriebnahme und Integration sind alles Bereiche, die in einem IoT-System berücksichtigt, dokumentiert und klar gelöst werden müssen. Darüber hinaus müssen diese Aspekte neu bewertet werden, wenn dem System über Produkteinführung, -weiterentwicklung oder Erweiterung des Ecosystems neue Elemente hinzugefügt werden.

Continuous Engineering und agile Methoden

Continuous Engineering ist kein einzelner Prozess. Es ist eine Denkweise, welche die Anwendung detaillierter Design- und Entwicklungsprozesse lenkt. Daher kann Continuous Engineering in Einklang mit agilen Methoden implementiert werden, eine Kombination, die sich gut eignet für die IoT-Entwicklung.

Agile Methoden sind in vielen Bereichen der Softwareentwicklung gut etabliert: Sie liefern beständig hohe Produktivität, Qualität sowie Fokus auf das Design und verbessern gleichzeitig die Reaktionsfähigkeit bei Änderungen. Diese Eigenschaften sind für die Entwicklung von IoT-Software von entscheidender Bedeutung, weshalb agile Methoden sehr attraktiv sind für Unternehmen, die für das IoT entwickeln.

In den letzten Jahren setzte sich die Erkenntnis durch, dass agile Methoden nicht nur für die Softwareentwicklung relevant sind, sondern mehr und mehr auch für Systems-Engineering-Aktivitäten.

Agile Methoden bieten eine Reihe von Vorteilen:

- Da die Zusammenarbeit mit Kunden priorisiert wird, können agile Methoden dazu beitragen, dass die passende Lösung geliefert wird.
- Durch die dynamische Verwaltung von Entwicklungsprioritäten können agile Methoden helfen, das Projektergebnis im Auge zu behalten, falls es große Unsicherheiten bezüglich der Ausrichtung der korrekten Lösung in einem Projekt gibt. Dies ist eine bedeutsame Möglichkeit bei der IoT-Entwicklung.
- Durch die Vermeidung technischer Lasten und die ausschließliche Lieferung von getesteten und funktionierenden Produkten können mit agilen Methoden Systemintegrationsprobleme vermieden werden, die zum Scheitern eines Projekts führen könnten.
- Durch den Wegfall unnötiger Arbeiten garantieren agile Methoden eine effiziente Projektdurchführung.

Agile Techniken können für den Einsatz in der IoT-Entwicklung angepasst werden, müssen dafür aber sorgfältig ausgewählt und angewendet werden. Einige agile Techniken wurden ursprünglich für kleine, unabhängige Teams entwickelt, die an einem Ort gemeinsam an Projekten arbeiteten, bei denen von einer beschränkten Lebenszeit ausgegangen wurde. Derartige Techniken können das Ausfallrisiko eines IoT-Projekts erhöhen. Andere Techniken und Werkzeuge sind hingegen besser für den Einsatz in IoT-Projekten geeignet. So stellt beispielsweise das Scaled Agile Framework (SAFe) getestete Mechanismen zur Minimierung der Risiken bei der Anwendung und Skalierung agiler Methoden für IoT-Projekte bereit. Ab SAFe v4.0 unterstützt das Framework agiles Systems Engineering.

Die IBM Internet of Things Continuous Engineering-Lösung

Die IBM IoT Continuous Engineering-Lösung bietet eine gemeinsame Entwicklungslösung für Systemingenieure, Entwickler von Embedded Software und Entwickler von IoT-Anwendungssoftware. Integrierte Werkzeuge für die Datenvisualisierung, Organisation, Analysen und Berichte unterstützen Benutzer dabei, komplexe Engineering-Daten zu deuten und zu teilen.

Danke der Unterstützung von Open Services for Lifecycle Collaboration (OSLC) können weitere Funktionen von IBM und Drittanbietern hinzugefügt werden. Die Lösung beinhaltet auch eine anpassbare Prozessunterstützung mit vorkonfigurierten Prozessinhalten für Systems Engineering- und Softwareteams, agilen Support (SAFe) und zahlreiche Industriestandards.

Die Schlüsselkompetenzen der Lösung entsprechen den Anforderungen von Systemtechnikern, Entwicklern von Embedded Software und Entwicklern von IoT-Anwendungssoftware, wie in Tabelle 1 dargestellt.

Tabelle 1. Wichtige Funktionen der IBM IoT Continuous Engineering-Lösung und deren Anwendbarkeit auf drei IoT-Engineering-Aktivitäten (mehrere Häkchen verweisen auf mehrere Einsatzmöglichkeiten)

Funktionen	Systems Engineering	Entwicklung von Embedded Software	Entwicklung von IoT-Anwendungssoftware
Zusammenarbeit, Workflow-Planung und Management	✓ ✓	✓ ✓	✓ ✓
Anforderungsmanagement und Rückverfolgbarkeit	✓ ✓	✓	✓
Änderungs- und Konfigurationsmanagement		✓ ✓	✓ ✓
Modellbasiertes Systems Engineering	✓ ✓	✓	
Systemqualitätsmanagement	✓ ✓	✓ ✓	✓ ✓
Lebenszyklus-Informationsmanagement	✓ ✓	✓ ✓	✓
Agile Planung und Management	✓	✓ ✓	✓ ✓
Release- und Bereitstellungsmanagement		✓ ✓	✓ ✓
Kontinuierliche Rückmeldung	✓ ✓	✓ ✓	✓ ✓

Wenn die Lösung in Ihrem Unternehmen eingesetzt wird, kann jede Engineering-Disziplin ihre erforderlichen Engineering-Verfahren mithilfe von Prozessen durchführen, die genau dafür entwickelt wurden. Aufgrund der gemeinsamen Nutzung der zugrundeliegenden Werkzeuge können die Ingenieure aus den verschiedenen Disziplinen darüber hinaus leichter zusammenarbeiten und Informationen austauschen. Das Ergebnis ist eine optimal vernetzte Entwicklungsumgebung.

Fazit

Für Anwender von IoT-Produkten und Unternehmen, die solche Produkte entwickeln, bietet IoT neue Möglichkeiten für Produktfunktionen und Wertschöpfung. Aber so, wie der Wert eines IoT-Systems mehr ist als die Summe der Teile, ist auch die mit der Entwicklung dieser Systeme verbundene Herausforderung mehr als die Summe der Engineering-Aktivitäten für die einzelnen Komponenten.

Die Komplexität und Interdependenzen, die IoT-Systeme charakterisieren, erfordern eine enge Zusammenarbeit zwischen den Engineering-Disziplinen bei der Planung und Umsetzung der Funktionen. Darüber hinaus passen IoT-Lösungen nicht in das konventionelle, aus Design/Fertigung/Verkauf bestehende Geschäftsmodell. Die sofortige Rückmeldung aus der Betriebsumgebung wie auch die beschleunigte Übernahme von Designänderungen im laufenden Betrieb erfordern jeweils äußerst responsive Engineering-Fähigkeiten.

... so wie der Wert der IoT-Systeme mehr ist als die Summe der Teile, ist auch die mit der Entwicklung dieser Systeme verbundene Herausforderung mehr als die Summe der Engineering-Aktivitäten für die einzelnen Komponenten

Fußnoten

¹ Worldwide Internet of Things Forecast Update, 2016-2020, IDC www.idc.com/getdoc.jsp?containerId=US40755516

² Nest Thermostat Glitch Leaves Users in the Cold, NY Times, 13. Januar 2016 www.nytimes.com/2016/01/14/fashion/nest-thermostat-glitch-battery-dies-software-freeze.html?_r=0

Die Schlüsselbereiche des Engineerings, die für die Auslieferung eines erfolgreichen IoT-Produkts miteinander verbunden werden müssen, sind das Systems Engineering, die Entwicklung von Embedded Software und die Entwicklung der IoT-Anwendungssoftware. Mit Continuous Engineering werden diese Aktivitäten durch agile, kollaborative Prozesse verbunden, die von gängigen Werkzeugen unterstützt werden. Das Ergebnis sind Automatisierung, Management von Engineering-Artefakten, Daten und Rückverfolgbarkeit sowie analysegestützte Engineering-Einblicke, die eine bessere Entscheidungsfindung ermöglichen.

Continuous Engineering ist kein fester Satz von Prozessen, sondern kann angepasst werden, um den Anforderungen verschiedener Industriestandards zu genügen und den Bedürfnissen von Teams gerecht zu werden, die mit unterschiedlichen Lieferzyklen arbeiten – teils innerhalb desselben Projekts.

Agile Ansätze sind wichtig, um das Engineering für das IoT zu optimieren. Jedoch müssen agile Methoden auf Umfang und Stringenz der IoT-Entwicklung abgestimmt werden. Mit dem Scaled Agile Framework können agile Continuous Engineering-Prozesse für die IoT-Lösungsentwicklung angepasst werden.

Die IBM IoT Continuous Engineering-Lösung stellt die Werkzeuge und agilen Prozesse bereit, um ein skalierbares, Umfeld ohne Reibungsverluste zu schaffen, in dem Systemingenieure, Entwickler von Embedded Software und Entwickler von IoT-Anwendungssoftware zusammenarbeiten und eine wirtschaftlich sinnvolle Balance finden können zwischen Verlässlichkeit einerseits sowie Geschwindigkeit und Kosten der Bereitstellung andererseits.

Weitere Informationen

Wenn Sie mehr über IBM Lösungen für IoT Engineering erfahren möchten, wenden Sie sich bitte an den zuständigen IBM Vertriebsbeauftragten oder IBM Business Partner, oder besuchen Sie uns unter: ibm.co/eng5

IBM Deutschland GmbH
IBM-Allee 1
71139 Ehningen
Germany
ibm.com/de

IBM Österreich
Obere Donaustrasse 95
1020 Wien
ibm.com/at

IBM Schweiz
Vulkanstrasse 106
8010 Zürich
ibm.com/ch

IBM, das IBM Logo, ibm.com und Watson sind Marken der International Business Machines Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Website „Copyright and trademark information“ unter www.ibm.com/legal/copytrade.shtml

Dieses Dokument ist zum Datum seiner Erstveröffentlichung aktuell und kann jederzeit von IBM geändert werden. Nicht alle Angebote sind in allen Ländern verfügbar, in denen IBM tätig ist.

DIE INFORMATIONEN IN DIESEM DOKUMENT WERDEN AUF DER GRUNDLAGE DES GEGENWÄRTIGEN ZUSTANDS (AUF „AS-IS“-BASIS) OHNE JEGLICHE AUSDRÜCKLICHE ODER STILLSCHWEIGENDE GEWÄHRLEISTUNG ZUR VERFÜGUNG GESTELLT, EINSCHLIESSLICH, ABER NICHT BESCHRÄNKT AUF DIE GEWÄHRLEISTUNGEN FÜR DIE HANDELSÜBLICHKEIT, DIE VERWENDUNGSFÄHIGKEIT FÜR EINEN BESTIMMTEN ZWECK ODER DIE FREIHEIT VON RECHTEN DRITTER.

Gegenstand und Umfang der Leistungen bestimmen sich ausschließlich nach den jeweiligen Verträgen.

Erklärung zu bewährten Sicherheitsverfahren:
Zur Sicherheit von IT-Systemen gehört der Schutz von Systemen und Informationen durch Prävention, Erkennung und Gegenmaßnahmen bei unsachgemäßem Zugriff innerhalb und außerhalb des Unternehmens. Der unsachgemäße Zugriff kann dazu führen, dass Informationen verändert, zerstört, unterschlagen oder missbräuchlich verwendet werden bzw. zu Beschädigungen oder missbräuchlicher Verwendung Ihrer Systeme, einschließlich der Nutzung bei Hackerangriffen auf andere Systeme.

Kein IT-System oder -Produkt sollte als vollständig sicher betrachtet werden und kein Produkt oder keine Sicherheitsmaßnahme kann allein die unsachgemäße Verwendung oder den Zugriff vollständig verhindern. IBM Systeme und Produkte wurden als Teil eines umfassenden Sicherheitskonzepts entwickelt. Für die effektivste Umsetzung dieses Konzepts müssen notwendigerweise weitere geschäftliche Verfahren und möglicherweise andere Systeme, Produkte oder Services einbezogen werden. IBM GARANTIERT NICHT, DASS DIE SYSTEME ODER PRODUKTE IMMUN GEGENÜBER DEM BÖSWILLIGEN ODER ILLEGALEN VERHALTEN DRITTER SIND.

© Copyright IBM Corporation 2016

WWW12373-DEDE-00

