



□ Leigh Williamson

(E-Mail: leighw@us.ibm.com)

ist IBM Distinguished Engineer und beteiligte sich seit 1988 im Labor Austin, Texas an großen Softwareprojekten wie IBM OS/2, DB2, AIX, OpenDoc, Java, Component Broker und WebSphere Application Server. In seiner derzeitigen Funktion als Mitglied des IBM Rational Software Chief Technology Officer Teams nimmt er Einfluss auf die strategische Entwicklung von IBM Rational und ist maßgeblich an der Lösungsdefinition für die Entwicklung mobiler Anwendungen beteiligt. Er hat einen Bachelor in Informatik der Nova Universität und einen Master-Abschluss in Computer Engineering der University of Texas in Austin.

Eine neue Epoche im Mobile Computing – IBM Mobile Enterprise Solution

Beschleunigt durch die vor Kurzem stattgefundene Firmenübernahme von Worklight steht IBM nun vor der Fertigstellung einer Plattform für mobile Anwendungen, die die Basis der IBM Mobile Enterprise-Strategie bildet. Dieser Artikel behandelt die drei Themen der IBM Mobile Enterprise-Strategie und untersucht die einzelnen Komponenten von Worklight. Außerdem wird beschrieben, wie die Collaborative Lifecycle Management-Funktionen von IBM Rational zu Projekten für die Entwicklung von mobilen Anwendungen passen. Hierzu gehören auch Informationen zum Entwickeln und Testen von speziellem Code für mobile Anwendungen.

Besondere Herausforderungen bei der Entwicklung von mobilen Anwendungen

In allen Branchen hat man mittlerweile realisiert, dass das Zielpublikum von Geschäftsanwendungen immer weniger traditionelle PCs wie Desktops und Laptops einsetzt, sondern stattdessen in verstärktem Maße mithilfe mobiler Geräte wie Smartphones und Tablet-Computern über das Internet auf die benötigten Informationen zugreift. Auf der ganzen Welt nutzen die Menschen heute mobile Geräte, die man überallhin mitnehmen kann und die benutzerfreundlicher und intuitiver zu bedienen sind, um Informationen zu suchen und Services über das Internet abzurufen.

Die Entwicklung von Anwendungen, die auf modernen mobilen Geräten wie Smartphones und Tablet-Computern laufen, ist mit besonderen Anforderungen und speziellen Herausforderungen verbunden.

1. Gerätemaße und Eingabetechnologie

Anzeige und Bedienfeld sind bei mobilen Anwendungen wesentlich kleiner und die Benutzerinteraktion funktioniert verglichen

mit klassischer Software anders. Ein weiterer Unterschied bei mobilen Anwendungen ist die veränderte Art der Eingabe. Mit den mobilen Geräten wurden tastaturunabhängige „Gesten“ als effiziente und gängige Eingabemethode eingeführt.

2. Gestaltung des Bedienungskomforts und der Benutzerinteraktion

Bei mobilen Anwendungen muss ein größeres Augenmerk auf Bedienungskomfort und Benutzerinteraktion gelegt werden. Umfangreiche Tastatureingaben müssen bei mobilen Anwendungen vermieden werden, damit bei der Nutzung wesentlich kleinerer Touch-Tastaturen keine Frustration eintritt.

Da die Umwelt für Ablenkung sorgt, kann sich der Benutzer für gewöhnlich auch nicht über längere Zeit auf die mobile Anwendung konzentrieren. Der Bedienungskomfort und die damit verbundenen Gestaltungsaspekte müssen bereits als Teil der Anforderungen für mobile Anwendungen formuliert, in späteren Entwicklungsstufen umgesetzt und in Tests geprüft werden, sodass die Benutzerinteraktion und die „Konsumierbarkeit“ der

Anwendung eine größtmögliche Akzeptanz finden.

3. Wahl der Implementierungstechnologie

Es gibt ein ganzes Spektrum an Implementierungsplattformen für mobile Anwendungen auf dem Markt. Eine Herausforderung für die Entwicklungsteams mobiler Anwendungen ist es, Stärken und Schwächen der unterschiedlichen Technologien zu erkennen und sich auf Basis der Projekt- und Anwendungsanforderungen zu entscheiden.

■ Native Implementierung einer Anwendung

Bei einer „nativen“ Implementierung wird die Anwendung mit der Programmiersprache und den Programmierschnittstellen (Application Programming Interface, kurz API), die vom mobilen Betriebssystem eines speziellen Gerätetyps zur Verfügung stehen, geschrieben. Da die verwendeten APIs systemnah und speziell für dieses Gerät entworfen sind, kann die Anwendung jede Funktion und jeden

Service dieses Gerätes in vollem Umfang nutzen. Native Implementierungen von mobilen Anwendungen sind nicht ohne großen Aufwand auf ein anderes Betriebssystem portierbar. Eine native Apple iOS-App muss praktisch neu geschrieben werden, wenn sie auf einem Android-Gerät laufen soll. Dadurch wird diese Methode zur Implementierung von mobilen Geschäftsanwendungen sehr kostspielig.

■ **Webanwendungen**

Mobile Anwendungen, die mit modernen Webtechnologien implementiert werden, unterstützen die größte Bandbreite an mobilen Geräten, da die Web-Browser-Unterstützung für JavaScript und HTML5 auf allen relevanten Plattformen vorhanden ist. Außerdem bietet das Webprogrammiermodell für die Implementierung von mobilen Anwendungen einen Vorteil für die Unternehmen, deren Entwickler bereits in den Sprachen und Techniken der Webanwendungsentwicklung erfahren sind. Der Nachteil der reinen Webanwendungsimplementierung ist, dass diese Anwendungen keinen direkten Zugriff auf Funktionen und Features haben, die direkt auf dem mobilen Gerät laufen, wie beispielsweise Kamera, Kontaktliste etc.

■ **Hybride Implementierung von mobilen Anwendungen**

Die hybride Implementierung von mobilen Anwendungen ist eine Art Kompromiss zwischen der reinen nativen Implementierung und der reinen Webimplementierung. Die mobilen Apps werden mit Programmiersprachen und -techniken entsprechend des Industriestandards wie beispielsweise HTML5 und JavaScript geschrieben. Die App wird jedoch dann in ein nativ installierbares Format verpackt, das über App-Store-Mechanismen des gewünschten mobilen Betriebssystems verteilt wird.

4. Erstellung und Auslieferung von mobilen Anwendungen

Aufgrund des Innovationsdrucks müssen mobile Anwendungen früh auf den Markt gebracht werden. Entwicklungsprojekte stehen daher oft unter extremem Zeitdruck. Zeiträumen, bei denen zwischen Projektstart und Auslieferung nur ein paar Monate liegen, sind nicht selten.

Der Druck, mobile Apps schnell auszuliefern zu müssen, führt bei den meisten Mobilprojekten zur Anwendung von agilen Entwicklungsmethoden. Ein wichtiges Element bei agilen Entwicklungsverfahren sind die kontinuierliche Integration und fortlaufende Builds. Von Entwicklern durchgeführte Änderungen der Anwendung müssen umgehend auf allen mobilen Betriebssystemen, die von der Anwendung unterstützt werden sollen, umgesetzt werden.

5. Test

Ein weiterer Bereich, in dem die Entwicklung von mobilen Anwendungen eine große Herausforderung darstellt, ist das Testen. Das Testen von mobilen Anwendungen, sofern es korrekt durchgeführt wird, ist verglichen mit eher traditionellen Anwendungen um ein Vielfaches komplexer und teurer.

Im Gegensatz zu traditionellen PC- und Webanwendungen ist die Bandbreite an potenziell unterstützten mobilen Geräten und Release-Levels weitaus größer. Hunderte oder sogar tausende Testkonfigurationen bestehend aus Kombinationen von Geräten, Releaseständen des mobilen Betriebssystems, Ländereinstellungen und Geräteausrichtungen sind keine Seltenheit bei Anwendungen für Mobilgeräte.

Der IBM-Ansatz

Der IBM-Ansatz bei mobilen Anwendungen ist die Bereitstellung einer Plattform, die alle für die Erstellung, die Ausführung und das Management von mobilen Anwendungen erforderlichen Funktionen umfasst. Die IBM Mobile

Enterprise-Plattform ist in drei Bereiche gegliedert:

■ **Mobile Anwendung erstellen, verbinden und ausführen**

Die Funktionen dieses Teils der Plattform umfassen die Tools, die für die Erstellung des Codes und die Verbindung der mobilen Anwendung mit den über das Internet verfügbaren Unternehmensservices und -daten erforderlich sind. Die mobilen Laufzeitkomponenten aus der Worklight-Übernahme sind Teil dieses Pakets und werden später im Einzelnen beschrieben.

■ **Mobile Anwendung verwalten und schützen**

Nach der Erstellung der mobilen Anwendung wird diese bereitgestellt und produktiv geschaltet. Ab diesem Punkt ist das Unternehmen für die Verfügbarkeit der Anwendung verantwortlich, muss Updates planen, ausliefern und sicherstellen, dass die Applikation vor Hackern geschützt bleibt.

■ **Mobile Anwendung erweitern und transformieren**

Die Kommunikation zwischen der mobilen Anwendung und den Unternehmensservices enthält wertvolle Informationen für das Unternehmen. Diese Informationen werden häufig für Analysen verwendet. Viele Unternehmen wollen außerdem Verbindungen zu Social-Networking-Funktionen in ihrer mobilen Anwendung schaffen.

Worklight Overview

	<p>Worklight Studio</p> <p>The most complete, extensible environment with maximum code reuse and per-device optimization</p>
	<p>Worklight Server</p> <p>Unified notifications, runtime skinning, version management, security, integration and delivery</p>
	<p>Worklight Runtime Components</p> <p>Extensive libraries and client APIs that expose and interface with native device functionality</p>
	<p>Worklight Console</p> <p>A web-based console for real-time analytics and control of your mobile apps and infrastructure</p>

Abb.: Worklight-Überblick, ©2012 IBM Corporation

Die IBM Mobile Enterprise-Software umfasst Lösungen für alle drei funktionalen Bereiche der Plattform. Sie enthält Entwicklungstools, Managementfunktionen und Laufzeitelemente, die alle zum IBM Angebot einer mobilen Plattform gehören.

Laufzeitumgebung für mobile Anwendungen mit der IBM Mobile Enterprise-Strategie

Einige Herausforderungen, die sich bei der Entwicklung von mobilen Anwendungen auf Unternehmensebene stellen, können nicht alleine von Entwicklungstools und -verfahren gemeistert werden. Um ein einziges gemeinsames Programmiermodell für mobile Anwendungen bereitstellen zu können, hat IBM das Unternehmen Worklight übernommen und plant nun, deren Modell als IBM Mobile Enterprise-Softwarelösung anzubieten.

Der wesentliche Wert von Worklight liegt in den Laufzeitkomponenten. Sie unterstützen eine einzige, auf Standards basierende Schnittstelle (Common Programming Interface) und bilden somit die Schlüsselkomponente der breiter angelegten mobilen IBM-Plattform.

Wie in der **Abbildung** dargestellt, reichen die Worklight-Funktionen von Entwicklungstools bis zur Managementkonsole.

IBM Rational Collaborative Lifecycle Management

Die umfassende IBM-Entwicklungslösung für mobile Geschäftsanwendungen kombiniert die leistungsstarken Funktionen der Anwendungsentwicklung im Team von IBM Rational Collaborative Lifecycle Management (CLM) mit den mobilen Tools und Laufzeitfunktionen der IBM Mobile Enterprise-Lösung. Diese Lösung für die Entwicklung von mobilen Geschäftsanwendungen bietet eine umfassende Managementplattform für den gesamten Softwarelebenszyklus. Sie unterstützt die Zusammenarbeit an gemeinsamen Aufgaben und verbindet die verschiedenen Artefakte miteinander, die im Verlauf des Produktlebenszyklus entwickelt werden.

Diese Lösung bietet außerdem Bereitstellungs-Workflows und Aufgabenmanagementfunktionen für die effektive Durchführung von Projekten zur Entwicklung von mobilen Anwendungen. Sie wird erweitert durch die Integration von mobile-spezifischen Funktionen, passend zur jeweiligen Projektphase.

Voller Lebenszyklus für das Projekt

Der Lebenszyklus eines Softwareentwicklungsprojektes folgt im Allgemeinen immer demselben Muster – unabhängig von der Art der Software, die gerade entwickelt wird. Er beginnt damit, dass basierend auf Analysen eine Geschäftsentscheidung für die Investition in die Anwendung getroffen wird. Die Anforderungen der Anwendung werden aufgelistet und ausgearbeitet. Diese Anwendungsanforderungen werden in einem agilen Projekt in User Stories bzw. Arbeitselemente zerlegt und in einem sogenannten Work Backlog organisiert. Von dort werden sie dann für die Ausarbeitung innerhalb von Iterationen oder Releases ausgewählt.

Teammitglieder in den unterschiedlichsten Rollen wählen die Arbeitselemente aus und führen die damit verbundenen Aufgaben mithilfe diverser Werkzeuge aus. Die fertigen Arbeitsergebnisse werden dann an das Projekt geliefert. Die sich daraus ergebende Anwendung wird getestet und auf Erfüllung der Anforderungen hin untersucht. Der Prozess im Detail und der Lebenszyklus der Softwareprojekte in einem Unternehmen sind natürlich üblicherweise auf dessen Ziele und Strategien zugeschnitten.

Dieser Lebenszyklus unterscheidet sich grundsätzlich für mobile Anwendungen nicht. Generell entstehen solche „Mobile Apps“ heute im Normalfall in kleinen Teams, in vorhandener Infrastruktur und in hohem Maße benutzerinteraktiv. Für agile Methoden ist das perfekt. Während sich die spezifischen Anforderungen für eine Mobile App wahrscheinlich von denen anderer Software unterscheiden, sind die Tools und Prozesse für das Sammeln, Ausarbeiten und Kommunizieren solcher Anforderungen identisch.

Des Weiteren sind die notwendige Verknüpfung zwischen den Anforderungen und den für die Implementierung dieser Anforderungen erforderlichen Codeänderungen bei der Entwicklung von mobilen und traditionellen Anwendungen gleich. Das heißt, der Projektablauf und die Notwendigkeit alle Artefakte aus den verschiedenen Phasen der Entwicklung in Beziehung setzen zu können (Traceability) sind bei der Entwicklung von mobilen Anwendungen gleich denen in klassischen Projekten.

Agile Tools für die gemeinsame Entwicklung im Team wie beispielsweise IBM Rational Team Concert (RTC) ermöglichen

die Definition von mehreren kurzen Iterationen innerhalb des Gesamtprojekts. Arbeitselemente können auf einfache Weise aus dem Projekt-Backlog in einen bestimmten Iterationsplan geschoben werden. Wenn die Entwickler Dateien innerhalb ihres Entwicklungstools für mobile Anwendungen bearbeiten, das in das Softwareversionskontrollsystem eingebunden ist, werden diese Änderungen automatisch den Arbeitselementen im Iterationsplan zugeordnet, sodass alle Änderungen und Erweiterungen auf die zugehörigen Arbeitselemente zurückgeführt werden können.

Integration mehrerer Tools

Es gibt, wenn überhaupt, nur wenige Softwareentwicklungsprojekte, bei denen man mit einem einzigen Entwicklungstool auskommt. Für die meisten Projekte ist eine Vielzahl an Tools verschiedener Anbieter notwendig, damit die Anforderungen einer bestimmten Rolle oder Aufgabe im gesamten Projektlebenszyklus erfüllt werden können.

Es ist beispielsweise denkbar, dass ein einzelner, an einer mobilen Anwendung arbeitender Softwareentwickler mit einem einfachen Codeerstellungstool, passend zur mobilen Zielplattform der Anwendung, auskommt. Dieses Tool verfügt aber möglicherweise nicht über die Funktionen, die die Zusammenarbeit und Koordination erleichtern, wenn ein Entwicklungsteam an der Erstellung einer Anwendung arbeitet.

Durch die Integration von Codeerstellungstools individueller Entwickler mit einer kompatiblen Plattform für die gemeinsame Entwicklung im Team können agile Teams die Effizienz deutlich steigern und qualitativ hochwertige Ergebnisse erzielen.

Die Notwendigkeit zur Zusammenarbeit im Team

Mobile Anwendungen werden normalerweise von einem kleinen Team von Entwicklern mit unterschiedlichem Know-how und Fachwissen erstellt. Ein typisches Team kann aus Entwicklern für grundsätzliche Geschäftsabläufe und Webservices, ein paar Entwicklern von Benutzerschnittstellen, einem Designer für das Benutzererlebnis, ein paar Testern und einem Teamleiter oder Manager bestehen.

Durch die üblicherweise eng gesteckten Zeitrahmen bis zur Auslieferung der mobilen Apps muss auch ein kleines Team

höchst effizient arbeiten. Jede Verzögerung, die aus einem Missverständnis oder Kommunikationsproblem zwischen den Teammitgliedern entsteht, kann den gesamten Auslieferungszeitplan sprengen.

Projekte für die Entwicklung mobiler Anwendungen, die mehrere verschiedene mobile Betriebssysteme unterstützen müssen, erfordern die gemeinsame Nutzung und Wiederverwendung von Code. Ein Entwickler hat sich möglicherweise auf Android und ein anderer auf iOS spezialisiert. Ein klares Verständnis der Arbeit, die von jedem Teammitglied in inhaltlicher und zeitlicher Hinsicht verlangt wird, ist unabdingbar. Projektanforderungen, Termine, Pläne etc. stimmen in diesem Fall zwischen den unterschiedlichen Entwicklungssträngen überein, wobei sich Quellcode, Tests und Builds voneinander unterscheiden.

Integriertes Änderungsmanagement und Softwareversionskontrolle

Damit Codeänderungen sowohl für die tägliche Arbeit als auch aus Gründen der Revisionsicherheit nachvollzogen werden können, müssen alle mit einem bestimmten Arbeitselement verbundenen Codeänderungen in einer Liste geänderter Quellcode-dateien, einem Change-Set, zusammengefasst werden. Das Set bzw. die Liste muss als Ganzes bereitgestellt werden, sodass die gesamte Codeänderung als eine Einheit nachvollzogen werden kann. Dieser Prozess der Zusammenstellung eines Änderungssets muss so unproblematisch wie möglich erfolgen, damit der Entwickler nicht abgelenkt wird und sich weiter auf die Logik konzentrieren kann.

Auch die Prozesse für Versionssteuerung und Zusammenfügen/Rollback der Quellcodeänderungen müssen automatisch und intuitiv vonstattengehen. Jedes Mal, wenn ein Entwickler seinen Arbeitskontext verlassen muss, um eine andere Aufgabe auszuführen, entsteht eine Unterbrechung und eine potenzielle Verzögerung im Entwicklungsprozess.

Die Notwendigkeit der Rückverfolgbarkeit im gesamten Projekt

Eine typische agile Herangehensweise bei der Softwareentwicklung ist die Definition mehrerer kurzer Iterationen, in denen jeweils ein kleines Set an Funktionalitäten implementiert und validiert wird. Eine typische agile Iteration dauert zwischen zwei und vier Wochen. Teamleiter und Team wählen bestimmte Arbeitselemente pro

Iteration aus und die Mitarbeiter selektieren diese situativ zur Bearbeitung. Die erzielten Fortschritte werden über die Arbeitselemente verfolgt.

Diese Art von „Projektgesamtansicht“ und durchgängiger Rückverfolgbarkeit ist für jede Art von Softwareentwicklungsprojekt extrem wichtig. Das gilt aber insbesondere für die Entwicklung von mobilen Anwendungen, bei der die Teams nach rigiden Zeitplänen arbeiten und agile Entwicklungsmethoden anwenden. Diese „schlanken“ Entwicklungsteams können sich den Zeitaufwand nicht leisten, manuell nachzuverfolgen, ob und wann eine bestimmte einzelne Anforderung verifiziert und umgesetzt wurde.

Ein umfassender Testansatz

Mehrschichtige mobile Anwendungen zu testen impliziert, dass unterschiedliche Verfahren zur Testdurchführung eingesetzt werden müssen, damit ein gemeinsames, qualitativ hochwertiges Gesamtergebnis entstehen kann. Rational Quality Manager ist ein Programm für die Verknüpfung und das Management von verschiedenen Testausführungs-Treibern, die für das Testen von mobilen Anwendungen erforderlich sind.

Wie bereits früher erwähnt, gibt es mehrere Methoden für das Testen und die Validierung mobiler Apps, die heute auf dem Markt sind. Die auf IBM Rational Best Practices beruhende Empfehlung ist, dass bei einem effektiven Entwicklungsprojekt alle anwendbaren Methoden (siehe unten) eingesetzt werden sollen, da jede einzelne ihre Stärken aufweist.

Eine einzige erschöpfende Antwort auf Fragen beim Testen von mobilen Anwendungen gibt es eben nicht und die verschiedenen Methoden schließen sich nicht gegenseitig aus. Die effektivste Teststrategie ist die, bei der alle Formen der Testausführung in ausgewogener Form angewendet werden und die Ergebnisse der einzelnen Methoden in eine umfassende Qualitätsbeurteilung eingehen.

Beispiele für von IBM unterstützte Methoden zum Testen von mobilen Anwendungen:

■ *Manuelle Tests*

Die Beobachtungen des Autors zeigen, dass manuelle Tests heute in der Branche die gängigste Methode beim Testen von mobilen Anwendungen

sind. Es ist jedoch auch die zeitaufwendigste, fehleranfälligste und kostspieligste Methode. Lösungen, bei denen die manuellen Testfälle organisiert, die Tester durch die Testausführung geleitet und die Testergebnisse gespeichert werden, können die mit solchen Tests verbundenen Kosten wesentlich reduzieren. IBM Rational Quality Manager (RQM) bietet solche Funktionen.

■ *Emulatoren und Simulatoren*

Emulatoren werden mit allen Kits für die native Entwicklung von mobilen Betriebssystemen ausgeliefert und Simulatoren sind aus verschiedenen Quellen verfügbar, einschließlich IBM Rational. IBM Rational bietet mobile Simulatoren als Teil der Entwicklungstools für die IBM Mobile Enterprise-Lösung.

■ *Virtualisierung von Protokollen, Isolierung von Anwendungsschichten*

Da viele mobile Anwendungen eine vielschichtige Architektur aufweisen, kann es zeit- und kostenaufwendig sein, eine Infrastruktur für die Unterstützung der Testausführung auf dem mobilen Gerät zu erstellen. Hohe Kosten und Verzögerungen bei der Implementierung können durch die Nutzung der Lösung IBM Rational Green Hat vermieden werden.

Damit können Testteams den Aufbau komplexer Middleware-Umgebungen zum Zweck des Tests von Code auf mobilen Geräten weitgehend vermeiden. Green Hat kann die Mittelschicht, Back-End-Services und Protokolle emulieren, damit man sich bei der Testausführung auf den Client der mobilen App, die auf dem Gerät selbst läuft, konzentrieren kann.

■ *Instrumentierung auf dem Gerät und/oder Einsatz von Agents*

Letztlich gibt es die Anforderung, manuelle funktionale Verifizierung durch eine Form von automatisiertem Test des auf dem mobilen Gerät ausgeführten Codes zu ersetzen. Hierfür gibt es eine Reihe von Methoden verschiedener Anbieter. Eine typische Methode ist, auf dem Gerät, auf dem der automatische Test durchgeführt werden soll, zusätzlichen Code zu platzieren. Dieser Code agiert als ein lokaler Agent auf dem Gerät, der automatisierte Benut-

zereingaben an die Anwendung übergibt. Dann wird das Verhalten der Anwendung nach dieser Eingabe ausgewertet. Diese Methode zur Automatisierung der mobilen Funktionstests ergänzt andere hier beschriebene Methoden und kann in Kombination mit diesen effektiv eingesetzt werden.

■ *Device Clouds*

Die für die Anschaffung, Implementierung und Verwaltung aller verschiedenen Kombinationen von mobilen Testgeräten anfallenden Kosten sind auch bei gut finanzierten Projekten nicht vertretbar. Eine Möglichkeit, mit diesem Problem umzugehen, ist der Einsatz einer „Device Cloud“-Testlösung. Durch die Verfügbarkeit virtueller Geräte in der Cloud können die Kosten für die Anschaffung einer großen Anzahl von verwendeten Gerätetypen reduziert werden.

IBM Rational bietet gemeinsam mit einer Vielzahl von Geschäftspartnern eine Integration der umfassenden mobilen Testmanagementlösung (RQM) mit Device Cloud-Lösungen an.

Fazit

Eine größer werdende Zahl von Unternehmen in allen Branchen hat erkannt, dass mobile Versionen ihrer Geschäftsanwendungen unerlässlich sind. Somit gibt es einen Bedarf an Methoden zur Entwicklung von mobilen Anwendungen auf Unternehmensebene. IBM hat einen solchen Ansatz entwickelt.

Der IBM-Ansatz bei der Entwicklung von mobilen Anwendungen stützt sich auf fünf Schlüsselthemen:

- Architekten müssen die Benutzererfahrung beim Einsatz von mobilen Anwendungen verbessern.

- Die Integration steht an erster Stelle im Hinblick auf verbesserte ökonomische Prozesse bei Projekten zur Entwicklung von mobilen Anwendungen.
- Die Rückverfolgbarkeit von Anforderungen zu Tests und deren Verifizierbarkeit muss gegeben sein.
- Es muss mit hochagilen Methoden gearbeitet werden.
- Suites für automatische Regressionstests müssen entwickelt werden, damit eine schnelle Implementierung und niedrige Kosten bei Änderungsprozessen sichergestellt werden können.

Trotz der erhöhten Anforderungen ist es durch diesen Ansatz möglich, spezielle mobile Tools und Technologien mit derselben Effizienz und Stringenz einzusetzen wie bei der klassischen Entwicklung von Unternehmensanwendungen. ■