

## 11 Überblick über Extensibility

Nachdem wir in den letzten Kapiteln den Fokus auf die Administration gesetzt haben, soll in diesem und in den nächsten Kapiteln die Anpassung von vRealize Automation der Hauptgegenstand sein.

### 11.1 Mögliche Anpassungen und Tools

In den meisten Projekten tritt ab einem gewissen Zeitpunkt der Fall auf, dass man nicht mehr alle Aufgabenstellungen mit der reinen Administrationsoberfläche lösen kann. Dies gilt auch für vRealize Automation. Mögliche Szenarien für Anpassungen und Erweiterungen gibt es viele.

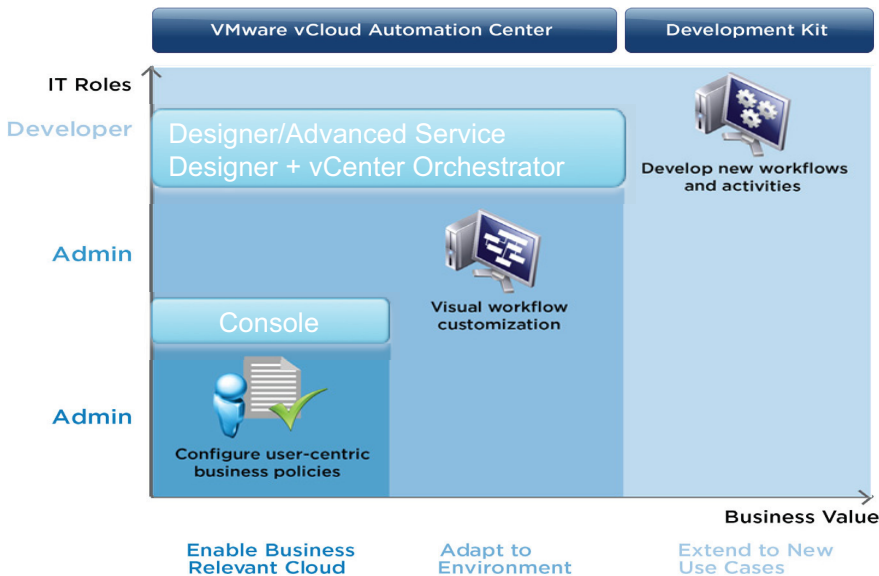
Einfache Erweiterungen umfassen oft die Anpassung vorhandener Workflows. So könnte beispielsweise ein Benutzer auswählen, welche Skripte nach der Provisionierung ausgeführt werden sollen, wie ein Rechnername lautet oder wo eine Maschine provisioniert werden soll. Vieles davon lässt sich administrativ mit Bordmitteln lösen. Im letzten Kapitel haben wir dazu bereits Custom Properties, Build Profiles und das Property Dictionary vorgestellt.

Oft kommen aber Anforderungen hinzu, die sich mit diesen Bordmitteln nicht mehr umsetzen lassen. Beispielsweise kann gefordert sein, vRealize Automation an die vorhandene IT-Infrastruktur bzw. deren Dienste anzubinden. Dies kann ein vorhandenes IP-Adressvergabe-System sein, ein System zur Vergabe von Rechnernamen oder auch eine Anbindung an ein betriebswirtschaftliches Auswertungssystem. In diesem Fall stehen der *vCenter Orchestrator* bzw. der *vCloud Automation Center Designer* zur Verfügung.

Ein anderes Szenario sind die Anything-as-a-Service-Dienste (XaaS). Der grundlegende Gedanke ist dabei, dass vRealize Automation für mehr als das Bereitstellen und Verwalten von Maschinen genutzt werden kann. Der mitgelieferte Service Catalog, das dahinterliegende Berechtigungssystem und die Anpassungsmöglichkeiten mittels vCenter Orchestrator gestatten die Bereitstellung von unterschiedlichsten Diensten, die eine IT-Abteilung ihren Kunden zugänglich machen will. Dies kann das Anlegen von Benutzern, das Ausführen von Jobs oder beliebige Aufgaben umfassen, die sich mittels Orchestrator automatisieren las-

sen. Die einzige Aufgabe seitens der IT-Abteilung besteht darin, diese Dienste mittels des *Advanced Service Designers* den Benutzern zugänglich zu machen.

Aber auch komplexe Anpassungen sind möglich. Mit dem *vCAC Cloud Development Kit* (CDK), das Sie zusätzlich erwerben können, lassen sich tiefgreifende Veränderungen in vRealize Automation implementieren. Dies erlaubt sogar die Erweiterung des internen vRealize-Automation-Modells um eigene Entitäten, die dann anschließend von allen Workflows angesprochen werden können. Auch ist es möglich, Workflows direkt in .NET zu implementieren. Abbildung 11–1 visualisiert diese Ansätze.



**Abb. 11–1** vRealize Automation-Erweiterungsmöglichkeiten

In diesem Kapitel geben wir Ihnen einen Überblick über diese Anpassungsmöglichkeiten. Eine detaillierte Besprechung erfolgt dann in den nächsten Kapiteln.

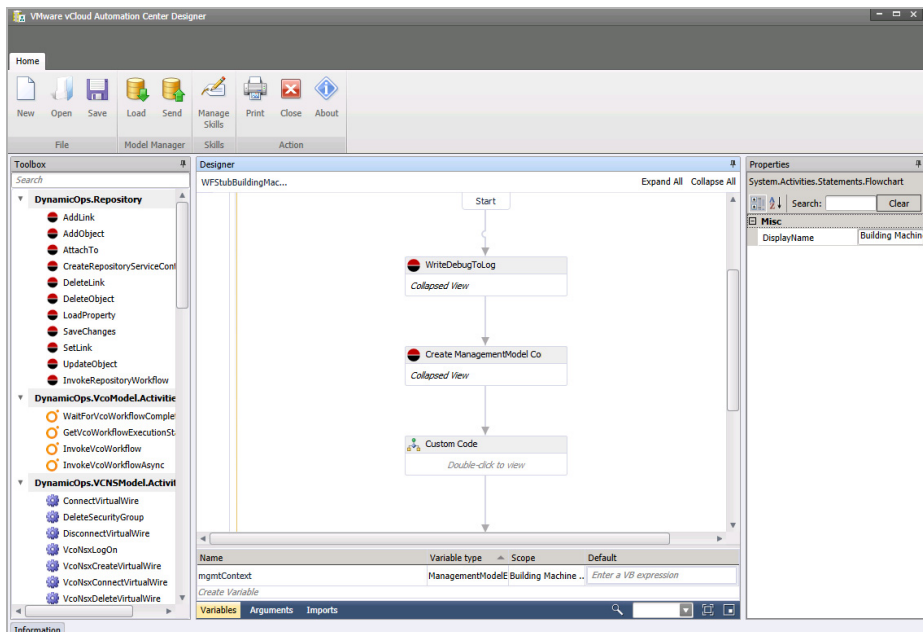
## 11.2 Anpassung mit vCAC Designer und VMware Orchestrator

Workflows lassen sich sowohl mit dem vCAC Designer als auch mit dem VMware Orchestrator anpassen. Der Grund dafür, dass zwei verschiedene Tools eingesetzt werden können, ist in der Historie zu finden.

vCAC Designer ist eine auf dem .NET-Framework basierende Windows-Anwendung. Ursprünglich ist vCAC Designer dazu entwickelt worden, Workflows anzupassen. Seine Stärken liegen darin, dass es sich als Windows-Applikation sehr performant verhält und die Bedienung der Anwendung intuitiv ist.

vRealize-Automation-Workflows basieren in der aktuellen .NET-Version auf der .NET-Bibliothek *Windows Workflow Foundation* (WWF). Dies begründet

eine weitere Stärke des vRealize Automation Designers: Es können vorhandene WWF-Workflows eingebunden werden. Dabei setzen die vRealize-Automation-Workflows direkt auf WWF auf. Somit werden sich .NET-Entwickler mit dem Tool sofort heimisch fühlen.



**Abb. 11–2** vCloud Automation Center Designer

Abbildung 11–2 zeigt das vCloud Automation Center Designer. Auf der linken Seite befindet sich die Toolbox, aus der Sie die Tasks mit Drag & Drop in den Hauptbereich bewegen. Workflows lassen sich dabei implementieren, indem Sie Tasks in den Hauptbereich ziehen und mittels Kanten miteinander verbinden.

Leider hat der vCAC Designer auch Schwächen. Aufgrund der Implementierung in .NET muss er auf einem Windows-Rechner installiert werden. Des Weiteren können zwar Workflows angepasst werden, eine richtige Programmierung ist aber nicht möglich. Es stehen zwar eine Reihe von Tasks (wie zum Beispiel für Kontrollfluss-Konstrukte, Deklarationen und Zuweisungen von Variablen) zur Verfügung, die deklarativ mit Drag & Drop zu einem Workflow kombiniert werden können, eine richtige Programmierung mittels einer Programmiersprache ist aber nicht möglich.

Ein weiterer Nachteil des Tools ist, dass ein Zugriff auf externe Systeme (z. B. Webdienste, Datenbanken) nur über Umwege, aber nicht nativ möglich ist. Dies kann nur geschehen, wenn ein externes Skript (z. B. Microsoft PowerShell) aufgerufen wird oder das vRealize-Automation-Modell mittels der CDK erweitert wird. Auch die Zukunft des Tools scheint nicht gesichert. Zwar basiert die aktu-