


Diese Leseprobe haben Sie beim
 edv-buchversand.de heruntergeladen.
Das Buch können Sie online in unserem
Shop bestellen.
[Hier zum Shop](#)

Einführung

Im SAP-Umfeld war Git lange Zeit nicht von Bedeutung. SAP hatte mit der SAP NetWeaver Development Infrastructure ein eigenes Produkt, das auf die Anforderungen an die Softwareerstellung und Versionierung im SAP-Umfeld ausgelegt war. Die SAP NetWeaver Development Infrastructure ist aber sehr schwergewichtig und in der modernen Entwicklung nicht mehr state-of-the-art. Git ist zwar kein vollwertiger Ersatz für die SAP NetWeaver Development Infrastructure, stellt jedoch im Bereich der Verwaltung und Versionierung von Entwicklungen eine leichtgewichtige Alternative dar. Git ist der De-facto-Standard der modernen Softwareentwicklung. In diesem Buch zeigen wir Ihnen, dass SAP und Git nicht widersprüchlich, sondern eine fast perfekte Kombination sind.

Aufbau des Buches

In **Kapitel 1**, »Einführung in Git«, erfahren Sie Wissenswertes über die Wurzeln von Git, die in der Linux-Entwicklung liegen. Die Linux-Entwicklung ist ein Best-Practices-Beispiel für eine global verteilte Entwicklung. Sie erfahren, was Git von anderen Quellcode-Management-Systemen unterscheidet und warum Git im SAP-Umfeld mittlerweile verstärkt eingesetzt wird. Es kann seine Stärken in der verteilten Entwicklung ausspielen und eignet sich auch für Continuous-Integration- und Continuous-Delivery-Projektmethoden, auf die in diesem Kapitel ebenfalls eingegangen wird.

In **Kapitel 2**, »Technische Grundlagen«, wird ausführlich auf die technischen Grundlagen von Git eingegangen. Git ist ein mächtiges Werkzeug, auf das Sie mit vielen grafischen Benutzeroberflächen zugreifen können. Doch auch die Verwendung dieser Benutzeroberflächen erfordert grundlegende Kenntnisse der dahinterliegenden Technologie. In diesem Kapitel erfahren Sie zunächst, wie das Git Command Line Interface (CLI) auf verschiedenen Plattformen installiert werden kann, um darüber Befehle abzusetzen. Im Anschluss erklären wir Ihnen das Zusammenspiel von lokalen und Remote-Repositorys und führen Sie in das Konzept der Branches ein. Es wird umfassend auf die wichtigen Git-Befehle `Commit`, `Merge`, `Stash` und `Rebase` eingegangen, die in der täglichen Arbeit mit Git permanent benötigt werden. Das Auflösen von Konflikten stellt Entwicklerinnen und Entwickler häufig vor Probleme. Daher möchten wir an dieser Stelle auch darauf ausführlich eingehen.

Grundlagen

In **Kapitel 3**, »Deployment-Optionen und Anbieter von Git-Clients«, beleuchten wir die verschiedenen Deployment-Optionen und stellen verschiedene Anbieter von Git-Clients vor. Ein Teaser vorweg: Git verfolgt einen agnostischen Ansatz, der unabhängig von den Produkten bestimmter Hersteller ist. Git kann z. B. lokal im Netzwerk des Entwicklers oder der Entwicklerin laufen, um ein On-Premise-Repository bereitzustellen. Es kann aber auch als Public Repository in der Cloud verwendet werden. Es gibt eine Vielzahl an Anbietern: Die im SAP-Kontext am weitesten verbreiteten Anbieter stellen wir Ihnen in diesem Kapitel vor. SAP spricht hier keine Empfehlung aus. Daher heben wir die Unterschiede heraus. Des Weiteren wird auf die mit den jeweiligen Tools ausgelieferten grafischen Clients eingegangen. Abschließend sehen Sie, wie ein On-Premise-Repository über den Cloud Connector sicher an die SAP BTP angebunden wird.

Einsatzszenarien

Da Sie nach der Lektüre der ersten drei Kapitel mit allen wichtigen Grundlagen ausgestattet sind, geht es in den folgenden Kapiteln an die unterschiedlichen Einsatzszenarien von Git im SAP-Kontext. In **Kapitel 4**, »Git in der SAPUI5-Entwicklung«, starten wir mit dem aktuell prominentesten Szenario, der SAPUI5- und der SAP-Fiori-Entwicklung. Die SAPUI5-Entwicklung kann in verschiedenen Entwicklerwerkzeugen (auch *Integrated Development Environment*, IDE) erfolgen. Jedes Werkzeug hat seine Vor- und Nachteile und bringt jeweils eigene integrierte Git-Clients mit. In diesem Kapitel erlernen Sie, wie Git in der Cloud auf Basis der SAP Web IDE und von SAP Business Application Studio angewendet wird. Auch die lokale Entwicklung in Visual Studio Code wird in diesem Kapitel beleuchtet. Damit sind Sie für alle Eventualitäten gerüstet.

Die Programmiersprache ABAP ist aus unserer Sicht einer der Erfolgsfaktoren von SAP. Für Git im ABAP-Kontext gab es aufgrund der in die ABAP Workbench integrierte Versionsverwaltung lange keine Daseinsberechtigung. Doch auch an dieser Front hat sich mit abapGit einiges getan. In **Kapitel 5**, »Git in der ABAP-Entwicklung«, stellen wir Ihnen abapGit, einen Git-Client für ABAP vor. Sie lernen hier, wie die Installation und Konfiguration erfolgt. Danach gehen wir auf die Verwendung von abapGit aus Eclipse auf Basis der ABAP Development Tools für Eclipse ein. Eclipse eignet sich für die klassische Entwicklung von On-Premise-Anwendungen auf Basis des SAP GUI, aber auch ABAP-Cloud-Entwicklerinnen und -Entwickler können Eclipse für ihre Entwicklungen auf Basis des ABAP-RESTful-Programmiermodells verwenden.

SAP bietet mit der SAP Business Technology Platform (SAP BTP) ein Platform-as-a-Service-Angebot in der Cloud an. Für Cloud-Entwicklungen wird neben ABAP auch massiv auf das SAP Cloud Application Programming Mo-

del gesetzt. Hierbei handelt es sich um ein mächtiges Programmiermodell, mit dem der Aufwand in der Entwicklung, verglichen mit anderen Technologien und Frameworks, massiv reduziert wird. In **Kapitel 6**, »Git in der Cloud-Entwicklung«, erhalten Sie eine kompakte Einführung in das SAP Cloud Application Programming Model. Danach widmen wir uns den Entwicklungswerkzeugen: Zuerst wird SAP Business Application Studio im Kontext des SAP Cloud Application Programming Models beleuchtet. Danach gehen wir auf die lokale Entwicklung in Visual Studio Code ein. An dieser Stelle gibt es einige Überschneidungen mit Kapitel 4, Sie werden hier aber zusätzliche Kniffe rund um die Verwendung der Git-Clients kennenlernen, die zuvor nicht abgedeckt wurden.

SAP HANA ist die Grundlage für die Zukunft von SAP. Dies gilt sowohl für die Cloud als auch für On-Premise-Landschaften. Auch in der SAP-HANA-Entwicklung führt kein Weg mehr an Git vorbei. Daher stellen wir in **Kapitel 7**, »Git in der SAP-HANA-Entwicklung«, Git im Kontext der SAP-HANA-Entwicklung vor. Die Einsatzmöglichkeiten von Git liegen üblicherweise in der Entwicklung nativer Anwendungen auf der Basis von SAP HANA. Die Entwicklung findet in der On-Premise-Landschaft in der SAP Web IDE bzw. in der Cloud in SAP Business Application Studio statt.

In **Kapitel 8**, »Continuous Delivery und Continuous Integration«, stellen wir mit Continuous Delivery und Continuous Integration (CI/CD) den »heiligen Gral« der modernen und agilen Softwareentwicklung vor. Bei *Continuous Delivery* und *Continuous Integration* handelt es sich um eine Methode, um Kunden oder Auftraggebern in regelmäßigen Intervallen Entwicklungsergebnisse bereitzustellen und alle Phasen der Anwendungsentwicklung zu automatisieren. Dabei spielt die Anwendung von Git eine entscheidende Rolle. In diesem Kapitel führen wir in die drei Hauptkonzepte Continuous Integration, Continuous Delivery und Continuous Deployment ein. Anhand von Beispielen lernen Sie, wie Sie CI/CD im SAP-Umfeld für die Entwicklung mit dem SAP Cloud Application Programming Model und für die SAP-Fiori-Entwicklung einsetzen können. Außerdem lernen Sie gängige Tools wie Apache Jenkins und Microsoft Azure DevOps sowie den Service SAP Continuous Integration and Delivery kennen.

Dieses Werk richtet sich an Entwickler*innen, Administrator*innen und Architekt*innen im SAP-Umfeld – sowohl in der klassischen On-Premise-Welt als auch in der Cloud. Entwickler*innen lernen die Werkzeuge für SAPUI5, SAP Fiori, SAP HANA, ABAP, RAP und CAP kennen; sie benötigen keinerlei Vorkenntnisse.

Wir empfehlen Ihnen, bei der Lektüre dieses Werkes mit Kapitel 1 und Kapitel 2 zu starten. Sollten Sie bereits Erfahrung in der Anwendung von Git ha-

CI/CD

Zielgruppe des Buches

ben, können Sie beide Kapitel überspringen. Die in Kapitel 3 vorgestellten Deployment-Optionen und Git-Clients sind für Entwickler*innen zwar »nice to know«, für die Verwendung von Git aber nicht zwingend erforderlich. Sollten Sie jedoch eine Administratoren- oder Architektenrolle einnehmen, führt für Sie kein Weg an Kapitel 3 vorbei. Wenn Sie ausschließlich an der Verwendung von Git in diversen SAP-Entwicklerwerkzeugen und Technologien interessiert sind, empfehlen wir Ihnen, abhängig von der gewünschten Technologie, Kapitel 4, Kapitel 5, Kapitel 6 und Kapitel 7 ausführlich zu lesen. Kapitel 8 dürfte allen voran Administrator*innen interessieren, die auf DevOps setzen und sich mit Continuous Delivery und Continuous Integration beschäftigen.

Informationskästen

Um Sie auf wichtige Informationen hinzuweisen und Ihnen so die Arbeit mit diesem Buch zu erleichtern, verwenden wir im Text die folgenden Symbole:



Kästen mit diesem Symbol geben Ihnen Empfehlungen zu Einstellungen oder Tipps aus der Berufspraxis.



Mit diesem Icon haben wir Warnungen oder mögliche Fallen gekennzeichnet.



Mit diesem Symbol haben wir zusätzliche, tiefer gehende Informationen gekennzeichnet.

Zusammenfassend haben wir versucht, ein möglichst rundes und vollständiges Bild der Verwendung von Git im SAP-Umfeld zu zeichnen. Wir danken Ihnen für das Interesse an unserem Buch und wünschen Ihnen viel Vergnügen bei der Lektüre.