

Agendavorschlag „Groovy“ Schulung

Mit dem Trainer soll vor der Erstellung des Trainings unsere Umgebung und vor allem die Idee dahinter (auch schon im Hinblick auf das noch kommende Orchestrierungstool und einen möglichen SelfService) besprochen werden, um die Gewichtung der einzelnen Elemente abzustimmen.

Punkte wie "Sourcecode Checkout" sind sicher hilfreich, aber durch unsere Templates eigentlich schon ganz gut abgefangen. Error Handling, Parallelisierung und Nebenläufigkeit wurden bei uns bisher nur "mit erhobenem Zeigefinger" erwähnt.

Als AddOn wäre eine „Geschichte von Groovy in Jenkins“ wünschenswert, also die Verwandtschaft zu Java, die Einschränkungen der Jenkins-Sandbox und wie man diese aufhebt/umgeht, die Möglichkeit, Klassen zu importieren etc.

Und wenn das nicht zu weit geht, also noch ein Stündchen Zeit ist, eine kurze Einführung in die Objektorientierung.

Die Jenkins Pipeline

- Features des Jenkins 2
- Groovy, Jenkins API und Plugins
- Wichtige Sprachmittel der Pipeline
- Pipeline Workspaces

Groovy Syntax

Viele Groovy-Schulungen (wie auch die Apache Groovy-Doku-Webseite) setzen Kenntnis von Java voraus und behandeln Java-Sprachmittel, die in Groovy wiederverwendet werden, nicht wirklich intensiv: Schleifen, Conditionals, etc. Das müsste bei uns auch kurz eingeführt werden.

- Vergleich zwischen Java und Groovy Syntax
 - Wichtige, für Groovy relevante Java-Sprachmittel
 - Was aus Java ist in Groovy uninteressant?
- Groovy Beans
- Strings
- Closures
- Kontrollstrukturen
- Optionale Typisierung
- Ranges
- Listen
- Maps
- Operatoren

Script Commands

- sh, bat, powershell
- Parameter und deren Anwendung
- Beispiel und Übung

Sourcecode Checkout

- SCM Anbindung in der Pipeline
- Git, SVN, TFS
- SCM Strategien und Jenkins
- Beispiel und Übung

Parameter & Flow control

- Pipeline Konfiguration, Trigger
- Pipeline Parameter
- Eingabe durch input
- Pipelines verketten

File Handling

- Wichtige und nützliche Sprachmittel
- Xml, yaml, json
- Austausch von Dateien
- Archivieren von Artefakten
- Beispiel und Übung

Error Handling

- Eigenschaft und Zustand einer Pipeline
- error und catchError
- try / catch / finally
- Best practices und Übung

Parallelisierung

- Das Schlüsselwort parallel
- Parallel im code oder als Variable
- Verteilung über Nodes
- Best practices und Übung

Nebenläufigkeit

- Motivation der Nebenläufigkeit

- lock und Lockable Ressources
- Milestones
- Beispiel und praktische Übung

Shared Library

- Arten von Libraries und deren Anwendung
- Libraries im Jenkins konfigurieren
- Libraries in der Pipeline verwenden
- Beispiel und best practices
- Praktische Übung

Pipeline-Architektur und best practices

- Pipelines in großen und komplexen Umgebungen
- Best practices und Zusammenfassung