

Jiemamy 開発プロセスガイド



version 0.2

製作著作 © 2009 Jiemamy Project and the Others

概要

Jiemamyプロジェクトが考える、Jiemamyをアプリケーション開発に適用する方法論を紹介します。

目次

まえがき	iii
1. このドキュメントのライセンス	iii
2. その他ライセンス	iii
1. はじめに	4
2. 構成管理	5
2.1. 構成管理の重要性	5
2.2. 管理の対象	5
2.3. 初期データも管理項目の一つ	5
3. Jiemamyを使用せず、DBの構成管理を行うケース	7
4. データベースの変更が発生した場合	8
5. Jiemamyを使用して、DBの構成管理を行うケース	9
6. おわりに	10
6.1. サポートが必要な場合	10
6.2. プログラム・ドキュメントにミスを発見した場合	10

まえがき

このチュートリアルでは、(TODO)

1. このドキュメントのライセンス

Copyright © 2006-2007 Jiemamy Project and the Others.

Licensed under the Apache License, Version 2.0 (the "License") you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

2. その他ライセンス

このドキュメントは、IPAフォント(バージョン 002.03)を使用しています。

IPAフォントは、一般利用者向けIPAフォント使用許諾契約に基づき、PDFに埋め込まれた形で再配布されています。 <http://ossipedia.ipa.go.jp/ipafont/>

第1章 はじめに

皆さんが関わっているWebアプリケーションプロジェクト。大抵はCVS, Subversion, Git, Mercurial等による構成管理が行われていると思います。このソフトウェアをチェックアウトしてから、ビルド・デプロイ等の手順を踏んで起動にこぎ着けるまで、どのようなステップがありますか？ 設定ファイルを環境に合わせて整備し、DBをインストールし、このSQLを流して…。

この、プロジェクトのビルド手順を標準化し、コマンドを1つ叩くだけで「今すぐ利用可能な状態」を作れるようにしよう、という試みの一つが Apache Mavenプロジェクトです。我々は、このビルド手法を「スマートビルド」と呼んでいますが、データベース(DB)の構築に関しては、理想的なスマートビルドを行う事が困難である感じました。

開発中にDBのスキーマが変更になる際、どのような手続きを踏みますか？ DB管理者と相談し、ER図を更新し、初期DBの初期化SQLを更新し、前バージョンのDBからの移行スクリプトを書き…。

このように、DBの管理・変更には大きなコストがかかります。

Jiemamyは、このDB構成の管理手順を標準化・補助することにより、DB状態の適切な管理を実現します。また、それにより、DBに対するリファクタリングや移行作業を簡潔に行うことができます。

第2章 構成管理

2.1. 構成管理の重要性

システム開発を行う上で、構成管理の必要性はもはや言うまでもないのですが、今一度、この構成管理の目的を挙げてみます。下記の他にも数多くの構成管理によるメリットはあると思いますが…。

- ・ 必要なファイル群をセットでまとめて入手できるようにする
- ・ プロジェクトの過去の状態をいつでも呼び出せるようにする
- ・ 誰がどこをどの様に変更したのかを記録する

ここでは、上に挙げた1つ目及び2つ目の役割について考えてみます。

2.2. 管理の対象

構成管理の主な対象はファイルとなりますが、どのようなファイルを構成管理すべきでしょうか。

プロジェクトによっては、テストコードはコミットしない、データファイル・設定ファイルはコミットしない、ドキュメントやDBのCREATE文はファイルサーバに置いて管理する、などの不思議なルールがまかり通っています。果たして、これらのファイルを構成管理下に置かないメリットはあるのでしょうか。我々は、そのデメリットの方が大きいと考えています。

- ・ テストを誰しもが実行できなくなる
- ・ 初めて開発環境を整える際に、既存メンバーの手を煩わせることになる
- ・ 過去のドキュメント・DBの状態を呼び戻せなくなる

その他にも色々なデメリットがあると思います。対して、これらのファイルを構成管理しながらない人が主張するのは「関係ないファイルを置きたくない」「置いても管理できない」等でしょう。ドキュメントは果たして「関係ないファイル」なのか。構成管理システム下に置かない事で「管理できるようになる」のか。今一度考えて頂きたいところです。

構成管理の理想的な運用目的は、「コンパイルできる状態のソースコードを維持」することではなく、「正常に動作する状態のシステムを維持管理する」ことです。この視点で考えると、例えばDBの初期化スクリプト(SQL文など)を構成管理しなかった場合、ソースコードについては過去の状態が呼び出せたとしても、それに依存するDBの状況が再現できません。当時のDBは、どのようなテーブルがあるべきだったのか。恐らくその点は、誰も管理していないと思います。つまり、昔の状態のアプリケーションを、もはや正常に動かす事ができないのです。

以上の理由により、我々は、ソースコードだけではなく、プログラムの動作に必要なファイルだけではなく、そのプロジェクトにまつわる全てのファイルを構成管理すべきだと主張します。

2.3. 初期データも管理項目の一つ

以上のように、アプリケーションの過去の状況を呼び出す為には、アプリケーションのソースコードだけではなく、データベースのスキーマも管理する 必要があることはお分かり頂けたかと思います。さらにもう一歩踏み込むと、例えば、初回ログインの為のアカウント情報や、デフォルトの設定情報など、テーブルはあったとしても空であって正常に動作しないアプリケーションも多く存在します。

従って、スキーマと同様、「初期データ」(初期状態でテーブルにINSERTされているべきデータ)も構成管理項目となります。

第3章 Jiemamyを使用せず、DBの構成管理を行うケース

(TODO)

第4章 データベースの変更が発生した場合

(TODO)

第5章 Jiemamyを使用して、DBの構成管理を行うケース

(TODO)

第6章 おわりに

6.1. サポートが必要な場合

配布ドキュメントやWebサイト上で解決できない問題に遭遇した場合は、以下のメーリングリストに質問を投稿することができます。

Jiemamy-usersメーリングリスト: jiemamy-users@lists.sourceforge.jp

質問を投稿する際は、以下のページを参考に投稿内容を検討してください。早期解決に繋がります。

技術系メーリングリストで質問するときのパターン・ランゲージ -- 「問題の解決」から「情報の共有」に至るために
<http://www.hyuki.com/writing/techask.html>

6.2. プログラム・ドキュメントにミスを発見した場合

プログラム・ドキュメント等にミスを発見した際も、上記メーリングリストに投稿してください。