

システム開発において、開発環境から検証環境への移行の手順および期間を教えて。対象システムはNTTデータIntramartです。プライベートクラウドからAWSへの移行になります。
因みに、データ移行は伴いません。

NTTデータのintra-mart（イントラマート）を使用したシステムにおいて、開発環境から検証環境への移行手順とその期間について、特に以下の条件下でのケースをご説明します：

- 対象：intra-martシステム
- 環境移行：プライベートクラウド → AWS（アプリケーション移行のみ／データ移行なし）

■ 移行手順（開発環境 → 検証環境）

以下は、一般的な intra-mart（Accel Platform）アプリケーションを、プライベートクラウド上の開発環境からAWS上の検証環境へ移行する手順です。

① AWS環境の準備（インフラ）

手順	内容
1.1	AWSアカウント／VPC／サブネット／セキュリティグループなどの基本設計・作成
1.2	EC2・RDS・EFSなど、必要なAWSリソースの構築（OS設定含む）
1.3	intra-mart用のWeb/AP/DBサーバの構築（パッケージ・ミドルウェア含む）
1.4	ロードバランサー（ALB）などの設定（必要に応じて）

 期間目安：1～2週間（環境構築・基本設計含む）

② intra-martアプリケーションの移行（WARファイル等）

手順	内容
2.1	intra-martのバージョン確認（検証環境に合わせる）
2.2	WARファイルや構成ファイル（imart.conf など）を開発環境から取得
2.3	AWS上の検証環境へデプロイ（アプリのみ、DBデータなし）

手順	内容
2.4	intra-martの起動・基本動作確認
 期間目安：1～3日	

③ 設定・ジョブ・スクリプトなどの適用（オプション）

手順	内容
3.1	intra-martの各種設定（IM-共通マスタ、IM-JOBなど）を手動／エクスポート／Git管理で反映
3.2	スクリプト（JavaScript、IM-Logic、画面カスタマイズ）の適用
3.3	認証方式や外部連携の設定確認
 期間目安：1～3日	

④ 動作確認・レビュー

手順	内容
4.1	intra-martログイン／機能単位の動作確認
4.2	簡易的なUAT（ユーザー受入テスト）
4.3	セキュリティ・ネットワーク（FW/ACL）も確認
 期間目安：2～5日	

■ 全体スケジュール（期間の目安）

フェーズ	期間目安
環境構築（AWS側）	1～2週間
アプリ移行・設定反映	1週間以内
総合テスト・レビュー	1週間程度

フェーズ

期間目安

合計

約2～3週間（最短10営業日程度）

⚠ 注意：AWS構築済み or AMIテンプレート化済みであれば、1週間以内で完了することも可能です。

■ 注意点

1. ライセンス対応

- intra-martのライセンスがAWS環境で動作するように設定が必要（ホスト名・MACアドレス変更に注意）。

2. 外部システム連携

- SAML認証、LDAP連携などがある場合、移行先でも再設定が必要。

3. SSL証明書／ドメイン

- https環境の再設定が必要なケースが多い。

4. ジョブ・スケジューラの時間設定

- 開発環境と検証環境でタイムゾーンや時間がずれると、バッチの誤動作リスクあり。

■ 補足：intra-mart特有の構成要素（確認対象）

- IM-共通マスタ
- IM-Logic
- IM-Workflow
- IM-JOB
- SAML認証／SSO設定
- カスタムJavaアプリケーション（im_contents）
- ポータルレイアウト／ガジェット

これらのうち、「開発環境で設定し、検証環境にも反映が必要なもの」は、Git管理やエクスポート機能（IM-共通マスタの設定エクスポート）を活用して再現性を確保するとよいです。

ご希望であれば、**チェックリスト形式**で「intra-martアプリ移行の確認項目一覧」もお作りします。必要ですか？

検証環境への移行後も、いくつか開発環境での開発をしたのちに検証環境へ移行する機能が発生する場合、すべての開発完了を待ってから検証環境へ移行すべきか？それとも、準備が完了した機能から順次検証環境へ移行すべきか？どちらが良いですか？

この問いは非常に重要で、intra-martに限らずすべてのシステム開発に共通する「リリース戦略（移行単位の考え方）」に関わる話です。以下に、それぞれの選択肢のメリット・デメリットを整理し、結論としてどちらを推奨するかをお伝えします。

■ 選択肢比較

① すべての開発が完了してから、まとめて検証環境に移行（バッチ移行方式）

項目	内容
✅ メリット	・検証環境の安定性が高く、整合性が保ちやすい ・検証の全体最適がしやすい ・リリース判定が一括でできる
⚠ デメリット	・一部の機能が完成していても、他の機能の遅延に引きずられて全体の検証が遅れる ・大規模変更の場合、問題発生時の切り分けが難しい

② 開発が完了した機能から、順次検証環境に移行（段階移行方式）

項目	内容
✅ メリット	・早期に検証ができるため、フィードバックが早く得られる ・リリースの柔軟性がある（アジャイル的） ・テスト観点が小さく、問題の特定がしやすい
⚠ デメリット	・移行頻度が増えるため、手順やリスクの管理コストが上がる ・依存関係がある機能の場合、単体では動作しない恐れあり ・バージョン管理／差分反映の仕組みが必要（特に intra-martでは注意）

■ 結論（どちらが良いか？）

✅ 原則として「段階移行」を推奨

ただし、以下の条件を満たす場合に限ります：

■ 段階移行を選ぶ条件（判断基準）

判断基準	YES/NO
intra-martアプリを 機能単位で分割／デプロイできるように作ってある	YES
intra-martの各機能が 他の機能に依存しない or 依存が管理されている	YES
GitやCI/CD、im-JOBのエクスポートなどにより、差分管理が確実にできる体制がある	YES
検証担当（またはユーザー）が、段階的な検証スケジュールに柔軟に対応できる	YES
検証環境が頻繁なデプロイに耐えうる構成（例：Blue/Green方式、バックアップあり等）	YES

 上記が「NO」なら、「まとめて移行（バッチ移行）」が無難

特に intra-mart では、以下のような特性から「全体整合性」が壊れやすいため注意が必要です：

- 設定情報（IM-共通マスタなど）がコードとは別管理
- ポータルレイアウトやワークフローなどが機能間で依存していることがある
- データベース初期値（マスタ情報など）が本番とずれると想定外動作する

■ 推奨する運用スタイル（折衷案）

ポイント	内容
1	週次または2週単位で「移行ウィンドウ」を設定（段階移行だが整理された運用）
2	機能ごとに JIRAなどで検証移行チケットを起票／管理（移行漏れ防止）
3	intra-mart構成ファイル（設定・スクリプト）をGitで管理し、タグ管理で差分移行
4	移行ログ／差分適用履歴を残す（後戻り可能に）

■ まとめ

質問

回答

すべての開発完了を待ってから検証環境へ移行すべきか？

✖ 推奨しません（納期や検証遅延リスクが高い）

完了した機能から順次検証環境へ移行すべきか？

✔ 推奨（ただし環境・体制に条件あり）

必要であれば、「段階移行のための運用チェックリスト」や「Git管理による intra-mart 反映フロー」も提供可能です。必要ですか？