



Sitecore CMS 6 およびそれ以降

Sitecore プロジェクトレビュー 推奨プラクティス

推奨プラクティスを使用してユーザビリティとパフォーマンスを最適化するためのデベロッパー ガイド

目次

第 1 章	イントロダクション	3
第 2 章	Sitecore 推奨プラクティス	4
2.1	テンプレートと ____ Standard Value	5
2.2	インフォメーション アーキテクチャ (コンテンツ構造)	7
2.3	セキュリティ (ユーザーおよびロール)	8
2.4	ワークフロー	9
2.5	メディア	10
2.6	キャッシュ	11
2.7	プレゼンテーション	12
2.8	ページ エディター	13
2.9	ソリューション コード	14
2.10	システム アーキテクチャ (ハードウェアおよびソフトウェア)	15

第 1 章

イントロダクション

この文書では、Sitecore プロフェッショナル サービスが Web サイト プロジェクト レビューを実施する際の推奨プラクティスについて説明します。この文書では、ユーザビリティとパフォーマンスに焦点を絞っています。

ここで説明する内容は、一般的な環境でのベスト プラクティスです。他のガイドラインと同様に、ここで説明する推奨事項を利用する前に実装要件の詳細を考慮する必要があります。

この文書は、Sitecore デベロッパーおよび管理者が最適な Web サイトを作成できるようにすることを目的としています。

この文書内の情報は Sitecore 6 およびそれ以降で利用可能です。

このマニュアルには次の章があります。

- **イントロダクション**
この文書の目的と内容を説明するイントロダクション (本章)。
- **Sitecore 推奨プラクティス**
取り扱う分野別にグループ分けした推奨ベスト プラクティスのリストを示します。

第 2 章

Sitecore 推奨プラクティス

以下の推奨プラクティスは、このテーマについてさまざまなブログから収集した情報に加えて、現場で Sitecore ソリューション レビューを実行した結果や Sitecore のマニュアルから収集したものです。

各セクションでは、Sitecore プロフェッショナル サービスの各担当者がどのようにサイトをレビューし、推奨プラクティスを適用する方法を調査し、その方法に沿って推奨事項を提案するかについて説明します。

この文書はすべての推奨プラクティスを完全には説明していません。今後改善できるように流動的な部分も残されています。

この章には次のセクションがあります。

- テンプレートと ____Standard
- インフォメーション アーキテクチャ (コンテンツ構造)
- セキュリティ (ユーザーおよびロール)
- ワークフロー
- メディア
- キャッシュ
- プレゼンテーション
- ページ エディター
- ソリューション コード
- システム アーキテクチャ (ハードウェアおよびソフトウェア)

2.1 テンプレートと __Standard Value

このセクションでは、テンプレートと __Standard Value に関する Sitecore 推奨プラクティスについて説明します。

- **命名規則** - テンプレート、フィールド、およびセクションには、単純で関連性のある、わかりやすい名前を使用します。デフォルトでは、ユーザーが入力した名前は技術系ユーザーと非技術系ユーザーの両方に対して表示されます。コンテンツ オーサーなどのビジネス ユーザーが簡単に認識して理解できる名前を選択します。
- **表示名** - アイテム名は半角英数文字に限られますが、表示名には日本語を使用できます。日本語の表示名を設定することでお客様に読みやすさを提供できます。
- **フォルダー構造** - フォルダー構造を作成し、Web サイト、サイトのセクション、セクションのコンテンツ、機能などで分類します。分類することで、特定のテンプレートに移動して簡単に見つけることができます。
- **継承の有効活用** - 独自のテンプレートに共通で使用するセクションやフィールドを配置し、さらに具体的なテンプレートでそのセクションやフィールドを継承できるようにします。たとえば、[ページ タイトルとテキスト] セクションの [タイトル] および [テキスト] フィールドは、複数のさまざまなコンテンツ テンプレートで使用されます。各コンテンツ テンプレートでこれらのフィールドを何度も使用せずに、単純に [ページ タイトルとテキスト] テンプレートを継承します。
- **フィールド名の重複を避ける** - フィールドが異なるセクションにある場合でも、同じテンプレート内でフィールド名を再利用しないようにします。継承チェーンに同じ名前を持つフィールドが 2 つ存在しないように注意します。明らかに同じ名前を持つフィールドがコンテンツ エディターの異なるセクションに 2 つ必要である場合は、フィールド定義アイテムの [タイトル] フィールドを使用します。このフィールドが設定されると、コンテンツ エディターは名前の代わりにこのフィールドを使用します。
- **テンプレートにアイコンを割り当てる** - 作成されるアイテムの種類に視覚的なヒントを提供します。
- **__Standard Value アイテム** - テンプレートにレイアウトの詳細、初期ワークフロー、および挿入オプションを定義します。この定義により管理作業が削減されます。システム設定を個々のアイテムで行うのではなく、まとめて管理します。
- **__Standard Value アイテム** - 意味のあるトークンを使用し、特定のフィールドにデフォルト値を入力する場合に期待される情報について、コンテンツ オーサーなどのビジネス ユーザーがヒントを得られるようにします。
- **__Standard Value アイテム** - \$name のようなトークンを使用し、新しいアイテムの作成時にコンテンツ オーサーが入力しなければならないテキストを減らします。
- **Image フィールド** - ソース フィールドを定義して、作成するアイテムに関連するメディア ライブラリ内の場所を示します。
- **リッチ テキスト エディターのフィールド** - リッチ テキスト エディターのフィールド内の HTML にスクリプトやスタイル情報、デバイス固有のマークアップを含めないでください。
- **リッチ テキスト エディターのフィールド** - デバイスにより使用できるマークアップが異なるため、極力使用を避け、マルチライン テキスト フィールドなどテキスト形式のフィールド タイプを使用し、装飾はすべてプレゼンテーション コンポーネントで行います。マルチ デバイスを考慮するサイトでリッチ テキスト エディターのフィールドを使用する場合は、、<I> タグ、フォント カラーやサイトコアの内部リンクのみに絞るなど、基礎的なマークアップの使用範囲に必ず限定します。

- **リッチ テキスト エディター**のフィールド - /sitecore/system/Settings/Html Editor Profiles の下の Core データベースにあるプロファイルにソース プロパティを設定し、ビジネス ユーザーが利用できるコントロールのみを表示します。
- /Sitecore/Templates/System の下にあるテンプレートを上書きまたは削除しないでください。
- フィールドに関する追加情報をユーザーに提供するには、個々のフィールド定義アイテムの [ヘルプを設定] オプションを使用します。また、定義アイテムの [**タイトル**] フィールドを使用して、ユーザーにフィールドの別の名前を表示することも検討します。
- テンプレートの名前は挿入オプションとして表示されることに注意してください。テンプレートには意味のある名前を付けるようにしてください。ユーザーにテンプレート名以外の異なるテキストを表示するには、[**表示名**] フィールドを使用します。

2.2 インフォメーション アーキテクチャ (コンテンツ構造)

このセクションでは、インフォメーション アーキテクチャに関する Sitecore 推奨プラクティスについて説明します。

- Web サイトの構造によく似たコンテンツ構造を作成します。これにより、ナビゲーション用コントロールを簡単に作成したり、セキュリティ ルールを使用してコンテンツ ツリーのさまざまな部分へのアクセス権を簡単に付与または拒否したり、コンテンツ構造内の特定のアイテムを見つける作業を簡単に行うことができます。
- パフォーマンスおよびユーザビリティを向上させるには、同じ親を共有する任意のノードの下のアイテム数を 100 アイテム以下に制限します。
- アイテムのバージョン数をできるだけ小さく制限します。サイトコアではアイテムのバージョン数を 10 以下にすることをお勧めしていますが、ポリシーにはこれよりも大きいバージョン数が指示されている場合があります。バージョンを管理するには、共有のソース モジュールである Version Manager (<http://trac.sitecore.net/VersionManager> から入手可能) を使用できます。
- コンテンツ オーサーなどのビジネス ユーザーの場合、パフォーマンスを向上させるには [スタンダードフィールド] をオフにします。
- 構造を定義する場合にリッチ テキスト エディターを使用しないようにします。これはプレゼンテーション層の役割です。
- ユーザーが適切なアイテムのみを選択したり、他の選択フィールドにある Sitecore クエリを選択できるようにするには、Treelist、DropTree、および TreelistEx の結果を絞り込む特別な構文を使用します。
- ホーム ノードとその子ノードのように非常に大きなツリーを表示する場合、または 1 つのアイテムにたくさんの Treelist フィールドがある場合は、Treelist ではなく TreelistEx を使用します。Treelist はレンダリングされるたびにツリーを計算しますが、TreelistEx は [編集] がクリックされた場合にのみツリーを計算します。
- デフォルトではツリー構造が URL に影響を与えることに注意してください。フォルダー内にアイテムを置くと、そのフォルダー名が URL の一部に含まれます。
- サイトのホーム アイテムの外側に URL を使用してアクセスするアイテムを配置したり、プレゼンテーションの詳細を設定しないでください。これを行うと、/Sitecore/content が URL に追加されます。
- 挿入オプションを適切に設定して、ツリーのさまざまな部分にユーザーが作成できるアイテムの種類を制限し、ツリーのトポロジを構成してください。これによりセキュリティを適用してさらに制限を強化できます。
- ルール (/sitecore/system/Settings/Rules/Insert Options) を使用して挿入オプションをさらに調整できます。これにより、動的な条件 (既存の子の数) やコンテンツの種類に応じて挿入オプションを追加または削除できます。この機能は、新しく "フォルダー テンプレート" を作成して挿入オプションを割り当てる必要をなくしたい場合に、特に便利です。

2.3 セキュリティ (ユーザーおよびロール)

このセクションでは、セキュリティ (ユーザーおよびロール) に関する Sitecore 推奨プラクティスについて説明します。

- **アクセス権を明示的に拒否するのではなく継承を解除する** - セキュリティ アカウントのアクセス権を明示的に拒否せずにアクセス権を拒否する場合の推奨プラクティスは、継承を解除することです。セキュリティ アカウントに対して明示的にアクセス権を拒否する場合、アクセス権の拒否を上書きする唯一の方法は、ユーザー アカウントに対して直接アクセス権を拒否することです。あるユーザーに、このロールおよびその他のロールを継承させて同様の適切なアクセス権を付与する場合は、これによってセキュリティ管理のオーバーヘッドが発生します。

例:

/home/contact us アイテムに対するロール A の書き込みアクセス権を拒否します。ロール B には同じアイテムに対する書き込みアクセス権を付与します。ユーザー AB にロール A とロール B を割り当てます。ただし、このアイテムに対する書き込みアクセス権が明示的に拒否されているため、ユーザー AB はこのアイテムに対する書き込みアクセス権を得ることができません。

セキュリティの継承を解除すると、権限にはデフォルトの拒否状態が設定されます。この設定は、いずれかのロールに明示的にアクセス権を割り当てることで上書きできます。アクセス権を明示的に拒否した場合は、アクセス権を明示的に付与しても上書きすることはできません。

セキュリティ ロールと継承の詳細については、http://sdn.sitecore.net/upload/sitecore6/japanese%20references/security_administrators_cookbook_sc62_jp.pdf のセクション 5.3 を参照してください。

- **ユーザーではなくロールにセキュリティを割り当てる** - ユーザー アカウントではなくロール セキュリティ アカウントにセキュリティ権限を設定することをお勧めします。この一意のアクセス レベルを持つユーザーがシステム内で 1 人のみである場合でも、ロール アカウントに権限を設定することをお勧めします。すべてのセキュリティ設定をロール セキュリティ アカウントに設定したセキュリティ システムは、管理が非常に簡単です。
- システムにログインしたユーザーに関連するコンテンツ ツリーの部分にアクセス権を制限します。たとえば、ニュース アイテムを担当するユーザーは [会社概要] ページへのアクセス権が不要である場合があります。このように制限することは、ユーザーが自分に関連する内容のみに焦点を絞ることができるだけでなく、コンテンツ ツリー全体をレンダリングする必要を無くすことでパフォーマンス面でのメリットも得られます。
- 個々のユーザーに関係のない機能を無効にして、リボン アイテムへのアクセス権を制限します。
- ユーザーがパスワードを空のままにしたり、わかりやすいパスワードを設定しないようにします。
- ユーザー プロパティのプロファイル設定を使用して、ログイン画面でユーザーがどのインターフェースを選択しても常に特定のインターフェースを使用するように指定します。
- ユーザーが必要な Sitecore Client ロールのみ所属するようにします。
- 管理ユーザー アカウントは管理タスク (主に他のユーザーのアイテムのロック解除) を実行する場合にのみ使用します。

2.4 ワークフロー

このセクションでは、ワークフローに関する Sitecore 推奨プラクティスについて説明します。

- ワークフローを有効にすることをお勧めします。ワークフローは、アイテムが公開可能になるまでに、一連の事前定義済みの状態を確実に通過するようにします。これにより、本番環境の Web サイトでコンテンツを公開する前に、コンテンツに対して常に適切なレビューと承認を実行することができます。
- 承認が不要な場合でも、ワークフローが非常に便利な場合があります。[パブリッシュ] ウィザードを実行したときに現在編集中のアイテムが公開されないようにするために、2 つの状態 (編集/パブリッシュ) のみの単純なワークフローが役立つ場合があります。さらに、これによって自動バージョン管理を実行できます。[パブリッシュ] 状態で自動公開アクションを使用すると、[パブリッシュ] ロールのメンバーでないユーザーもコンテンツを公開できます。
- 適切なワークフロー セキュリティを適用します。
- ワークフローの公開を特定のロールとユーザーに制限します。ワークフローで、デフォルトの最終手順を公開アクションにします。
- すべてのワークフロー状態を Email 通知で送信しないようにします。重要なワークフロー イベントに関して有用である場合にのみ、Email で通知します。または、Email の代わりに RSS フィードを使用できます。
- ワークフロー、状態の数、および関連するユーザーの数を最小限に抑えます。

2.5 メディア

このセクションでは、メディアに関する Sitecore 推奨プラクティスについて説明します。

- メディアはデータベースにストアすることをお勧めします。メディア資産の公開と展開が簡素化されるため、BLOB ベースのストレージ アプローチをお勧めします。
たとえば、ファイル システム ベースのメディアではサーバー間でディレクトリとファイルの同期が必要であるのに対して、データベースにストアされたメディアは本番環境のターゲットに簡単に公開できます。
- メディア ライブラリのフォルダー構造は、非技術系ユーザーでも簡単に移動できるように整理する必要があります。
- メディア アイテムの名前には、コンテンツ オーサーなどのビジネス ユーザーが簡単に認識して理解できる名前を付けます。
- メディア アイテムは Photoshop などのグラフィック プログラムを使用してレンダリングされるページのサイズを小さくし、パフォーマンスが向上するように最適化する必要があります。
- **Image フィールド** - ソース フィールドを定義して、作成するアイテムに関連するメディア ライブラリ内の場所を示します。

2.6 キャッシュ

このセクションでは、キャッシュに関する Sitecore 推奨プラクティスについて説明します。

- Sitecore データベースのキャッシュ (プリフェッチ、データ、およびアイテム) を設定および調整することをお勧めします。詳細については、http://sdn.sitecore.net/upload/sitecore6/japanese%20references/cache_configuration_reference-sc62-jp.pdf のセクション 4.2「キャッシュのサイズのチューニング」を参照してください。
- レンダリングされる出力、特に頻繁に取得される負荷の高いロジックおよびコンポーネントをキャッシュします。レンダリングのキャッシュ設定の詳細については、http://sdn.sitecore.net/upload/sitecore6/japanese%20references/presentation_component_reference_sc62_jp.pdf を参照してください。

2.7 プレゼンテーション

このセクションでは、プレゼンテーションに関する Sitecore 推奨プラクティスについて説明します。

- サブレイアウトのレイアウト数を制限します。デバイスごと、サイトごとに 1 つのレイアウトを持つことを目標とします。
- レイアウトの (マークアップの) 個々のページに静的に表示するコンポーネントを配置します。一部のページのみに表示するコンポーネントは、ブレースホルダーを使用して動的にバインドします。常に一緒に使用されるコンポーネントはサブレイアウトにグループ化します。
- 可能な場合は常にプレゼンテーションの詳細を __Standard Value に設定します。
- ページ エディター機能を有効にするには、FieldRenderer または他の Sitecore プレゼンテーション コントロールを使用します。disable-web-editing 属性が設定されている、編集可能にすべきではないフィールドをレンダリングする場合に、このコンポーネントを使用できます。
- 適切なイメージ操作パラメーターを使用して、サーバー上でイメージのサイズを変更し、大きなファイルが不必要にブラウザに送信されないようにします。
- XSLT で \$sc_currentitem ではなく \$sc_item を使用するか、Web サーバー コントロールで DataSource 属性を使用するか、サブレイアウトで適切なコードを使用して、DataSource を使用します。これにより、OMS で多変量テスト (Multi-Variant Tests、MVT) やパーソナリゼーション ルールを簡単に作成できます。
- プレゼンテーション コンポーネントのパラメーターを活用します。ユーザーがコンポーネントの動作 (リストに表示するアイテム数、使用する CSS クラス、表示するフィールドの URL など) を設定または変更できるようにするには、このパラメーターを使用します。
- プレゼンテーション コンポーネントの任意のパラメーターのテンプレートを作成し、パラメーター テンプレート プロパティを設定します。
- コンポーネントを適切にテストします。たとえば、新しいアイテムを持つメニューのレンダリングが正しく動作しても、大量のデータがある場合に問題が発生する可能性があります。

2.8 ページ エディター

このセクションでは、ページ エディターに関する Sitecore 推奨プラクティスについて説明します。

- サイトコアでは、非技術系ユーザーのためにページ エディターを有効にして設定することをお勧めしています。
- ユーザーがページに適切なコンポーネントを追加できるように、プレースホルダーを設定します。また、[編集] チェックボックスを使用して、ページ エディターのコンポーネントを特定のプレースホルダーに追加する機能を無効にします。セキュリティを適用して（書き込みアクセス権を無効にして）、特定のユーザーのみに対してプレースホルダーを無効にします。
- Sitecore 6.4 およびそれ以降では、[ページ エディター] ボタン、[データソースの場所]、および [データソース テンプレート] フィールドの使用も検討します。
- CSS を必ずテストします。ページ エディターの表示方法に CSS が影響を与える場合があります。

2.9 ソリューション コード

このセクションでは、ソリューション コードに関する Sitecore 推奨プラクティスの概要について説明します。

- サイトコアでは、C# の使用を強くお勧めしています。Sitecore のプログラミング例はすべて C# で作成されています。また、Sitecore サポートが使用する言語も C# です。
- Web、Windows、IIS、ASP.NET、XML、XSL、SQL、.NET、CSS、JavaScript、およびその他のテクノロジーに関するベスト プラクティスに従います。たとえば、コードにコメントを追加して、実装ではなく目的を記述します。
- パスまたは名前ではなく、可能な限り GUID を使用します。これは、コンテンツ ツリーの新しい場所にコンテンツを移動する場合に破損するのを防ぐだけでなく、パフォーマンスも向上します。
- クエリまたは高速クエリではなく、Sitecore 検索 API を使用します。これにより、コンテンツ構造ではなくインデックスに基づいてクエリを実行できます。検索の詳細については、<http://sitecoreblog.alexshyba.com/2011/02/8-reasons-to-use-new-search-in-sitecore.html> および <http://sdn.sitecore.net/Reference/Sitecore%206/Sitecore%20Search%20and%20Indexing.aspx> に投稿された Alex Shyba のブログを参照してください。
- XSLレンダリングで//、descendant軸、descendant-or-self軸を使用しないようにします。
- web.config ファイルを直接変更せずに、App_Config/Include ファイルを使用します。
- コードに情報をハードコードせず、*.config ファイルの <settings> を使用します。
- パイプラインまたはイベントを作成する場合は、パラメーターとプロパティを使用します。
- ラベルやその他の静的テキスト情報をハードコードせず、アイテムを使用したモデリングを検討するか、Sitecore で辞書を作成します。

2.10 システム アーキテクチャ (ハードウェアおよびソフトウェア)

このセクションでは、システム アーキテクチャ (ハードウェアおよびソフトウェア) に関する Sitecore 推奨プラクティスについて説明します。

サイトコアでは以下の内容を推奨しています。

- すべてのサーバー - Web およびデータベースを Windows Server 2008 R2 64 ビットで実行します。
- Web サーバーでは最新のセキュリティ更新プログラムを適用した IIS 7.x を実行します。
- IIS 7.x 統合モードを使用します。
- データベース サーバーでは最新のセキュリティ更新プログラムを適用した Microsoft SQL Server 2008 R2 64 ビットを実行します。
- Web サーバーとデータベース サーバーは別々のコンピューターで実行します。
- SQL Server 上の Core、Master、および Web データベースでインデックスを定期的に最適化するための保守計画を導入します。
- IIS - 静的コンテンツの圧縮を有効にします。
- IIS - CMS サーバーの動的コンテンツの圧縮を有効にします。
- IIS - 共通ヘッダー (具体的には /sitecore フォルダ) にコンテンツの有効期限ヘッダーが設定されていることを確認します。
- パフォーマンスの向上に関する追加情報については、SDN の「[Optimizing Performance in Sitecore](#)」の記事および以下のリファレンスを参照してください。
 - CMS パフォーマンス チューニング ガイド
<http://sdn.sitecore.net/Reference/References%20in%20Japanese/CMS%20Performance%20Tuning%20Guide.aspx>
 - OMS パフォーマンス チューニング ガイド
<http://sdn.sitecore.net/Reference/References%20in%20Japanese/OMS%20Performance%20Tuning%20Guide.aspx>