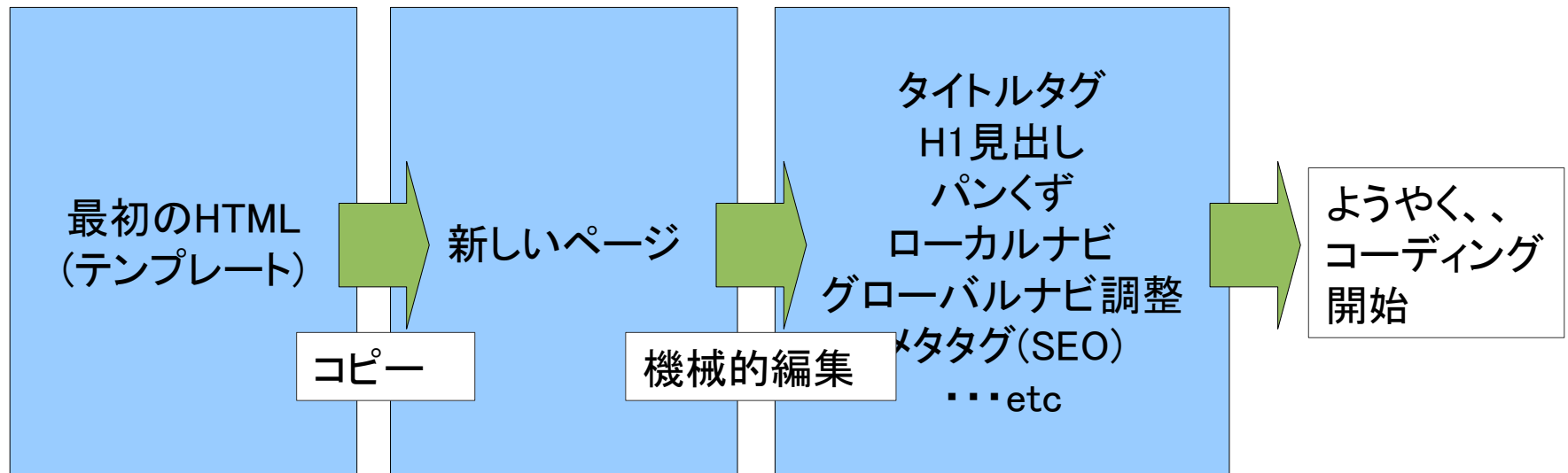

Pickles Framework のススメ

Version 2012/8/8

Introduction of “Pickles Framework”

背景 ～ 従来のコーディング手順には“機械的な作業”がいっぱい！

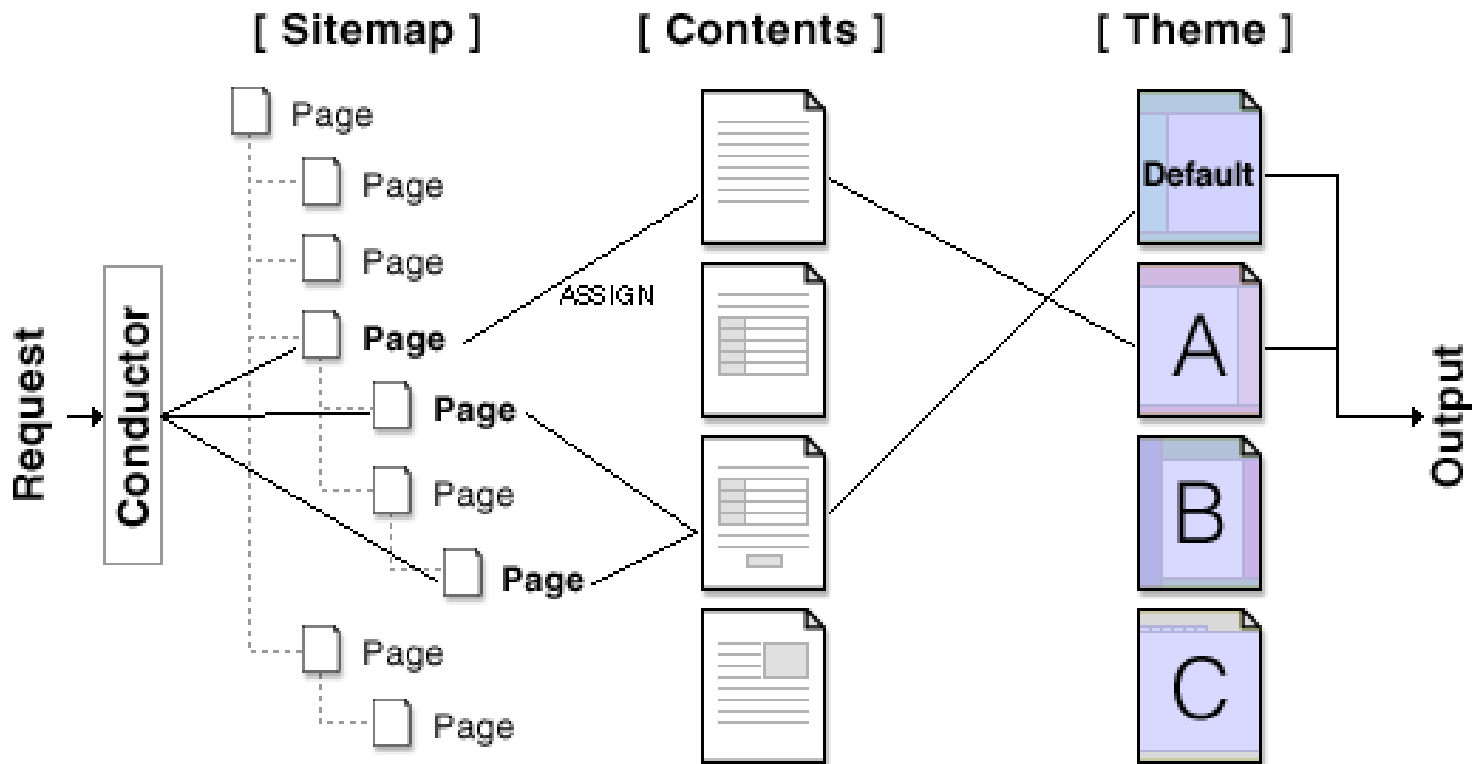


機械的な作業、意外と負担が大きい ←

- ・実は同じ情報を沢山コピーしている。(タイトルとか)
- ・ミスが起こりやすく、気づきにくい。
- ・確認するのも大変。

機械的な作業は、機械にやらせてもらおうじゃないか！

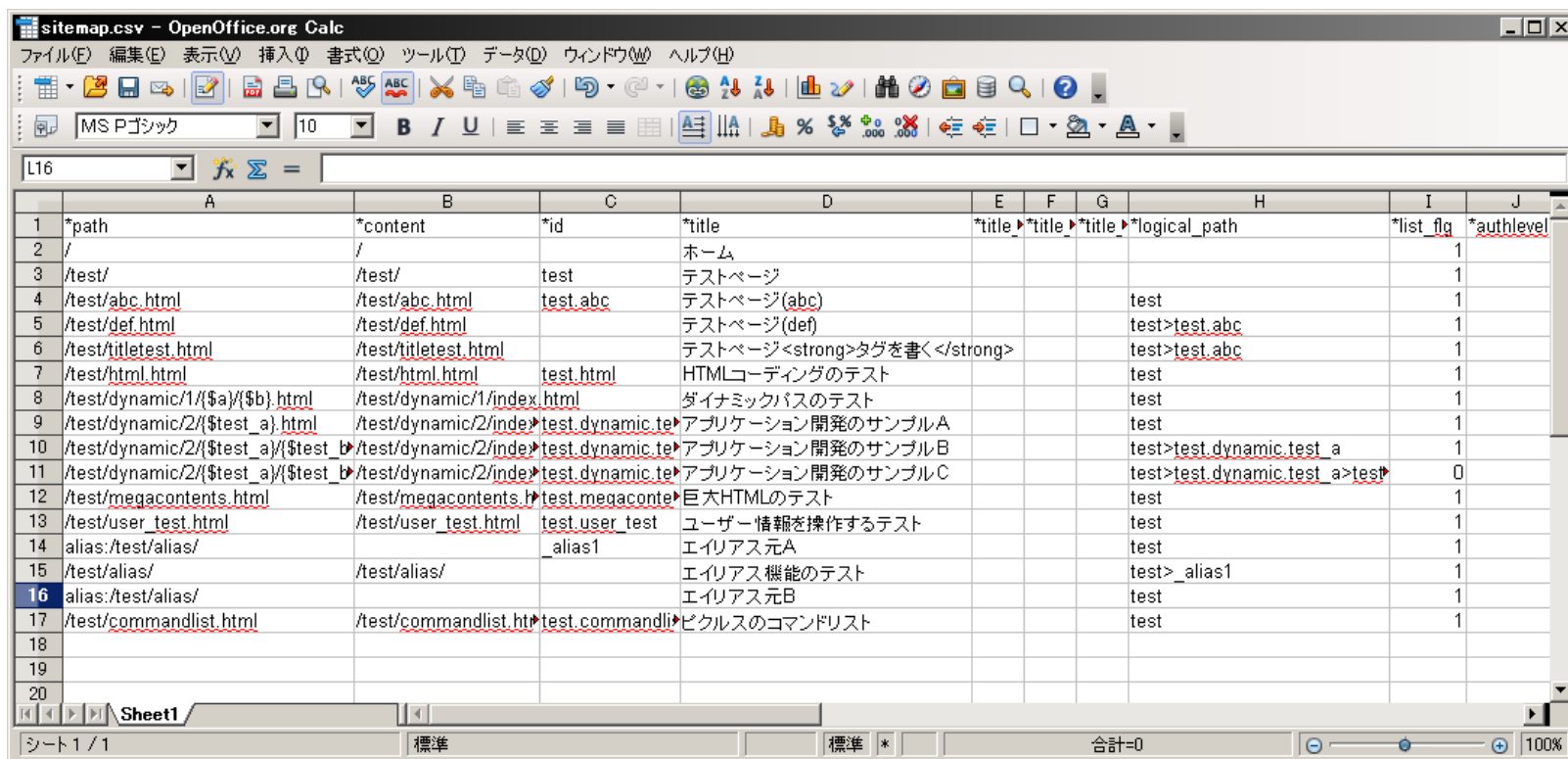
Pickles Framework の概念 : ウェブサイトの構造を分解して捉える。



一般的な「ウェブサイト」の構造を、
「サイトマップ」、「コンテンツ」、「テーマ」の3要素に
分解して捉え、整理します。

「サイトマップ」とは

「サイトマップ」は、CSV形式で管理された、
サイト全体の構造を示すページの一覧です。



	A	B	C	D	E	F	G	H	I	J
	*path	*content	*id	*title	*title	*title	*title	*logical_path	*list_flg	*authlevel
1	/	/		ホーム					1	
2	/test/	/test/	test	テストページ					1	
3	/test/abc.html	/test/abc.html	test.abc	テストページ(abc)				test	1	
4	/test/def.html	/test/def.html		テストページ(def)				test>test.abc	1	
5	/test/titletest.html	/test/titletest.html		テストページタグを書く				test>test.abc	1	
6	/test/html.html	/test/html.html	test.html	HTMLコーディングのテスト				test	1	
7	/test/dynamic/1/{%a}/{%b}.html	/test/dynamic/1/index.html		ダイナミックパスのテスト				test	1	
8	/test/dynamic/2/{%test_a}.html	/test/dynamic/2/index>test_dynamic.te		アプリケーション開発のサンプルA				test	1	
9	/test/dynamic/2/{%test_a}/{%test_b}.html	/test/dynamic/2/index>test_dynamic.te		アプリケーション開発のサンプルB				test>test_dynamic.test_a	1	
10	/test/dynamic/2/{%test_a}/{%test_b}/{%test_c}.html	/test/dynamic/2/index>test_dynamic.te		アプリケーション開発のサンプルC				test>test_dynamic.test_a>test_b	0	
11	/test/megacontents.html	/test/megacontents.html	test_megaconte	巨大HTMLのテスト				test	1	
12	/test/user_test.html	/test/user_test.html	test_user_test	ユーザー情報を操作するテスト				test	1	
13	alias:/test/alias/		_alias1	エイリアス元A				test	1	
14	/test/alias/	/test/alias/		エイリアス機能のテスト				test>_alias1	1	
15	alias:/test/alias/			エイリアス元B				test	1	
16	/test/commandlist.html	/test/commandlist.html	test.commandlist	ピクルスのコマンドリスト				test	1	
17										
18										
19										
20										

※キャプチャは、OpenOffice の Calc で開いた様子です。

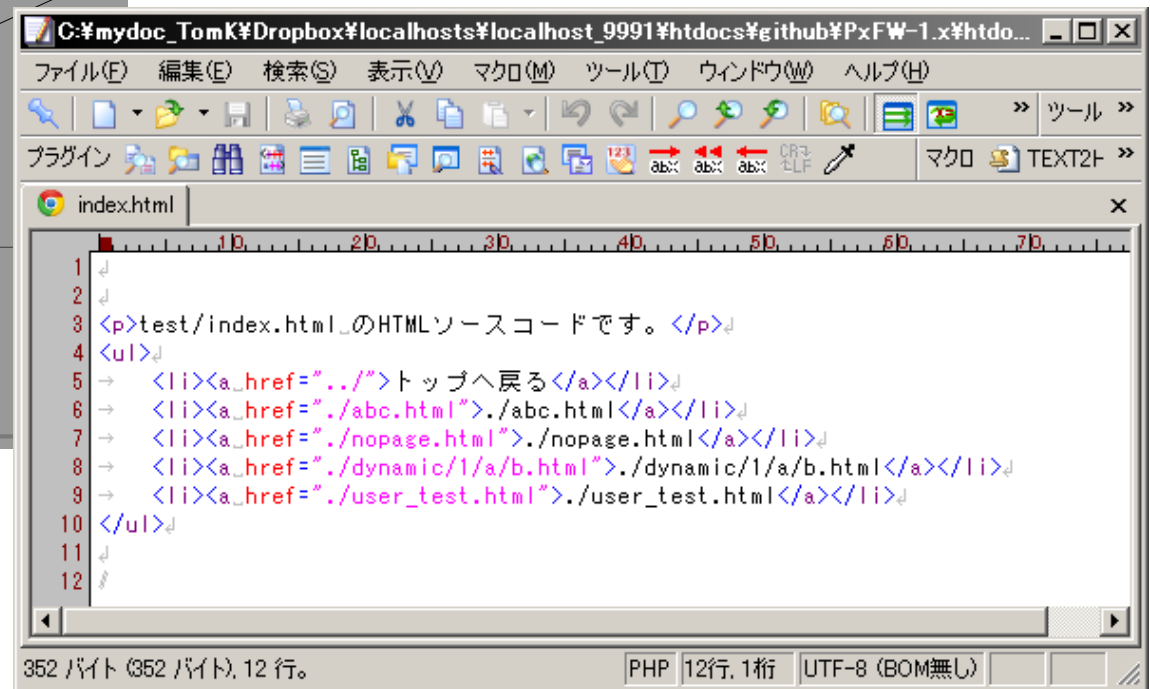
※Excel で作成した CSVファイルでは、日本語が正しく処理されない場合があります。

「コンテンツ」とは

「コンテンツ」とは、1枚のウェブページを構成するうちの、「内容の部分だけ」を指した概念です。



コンテンツは、この部分だけを記述する概念です。



↑ シンプルなHTMLによる実装

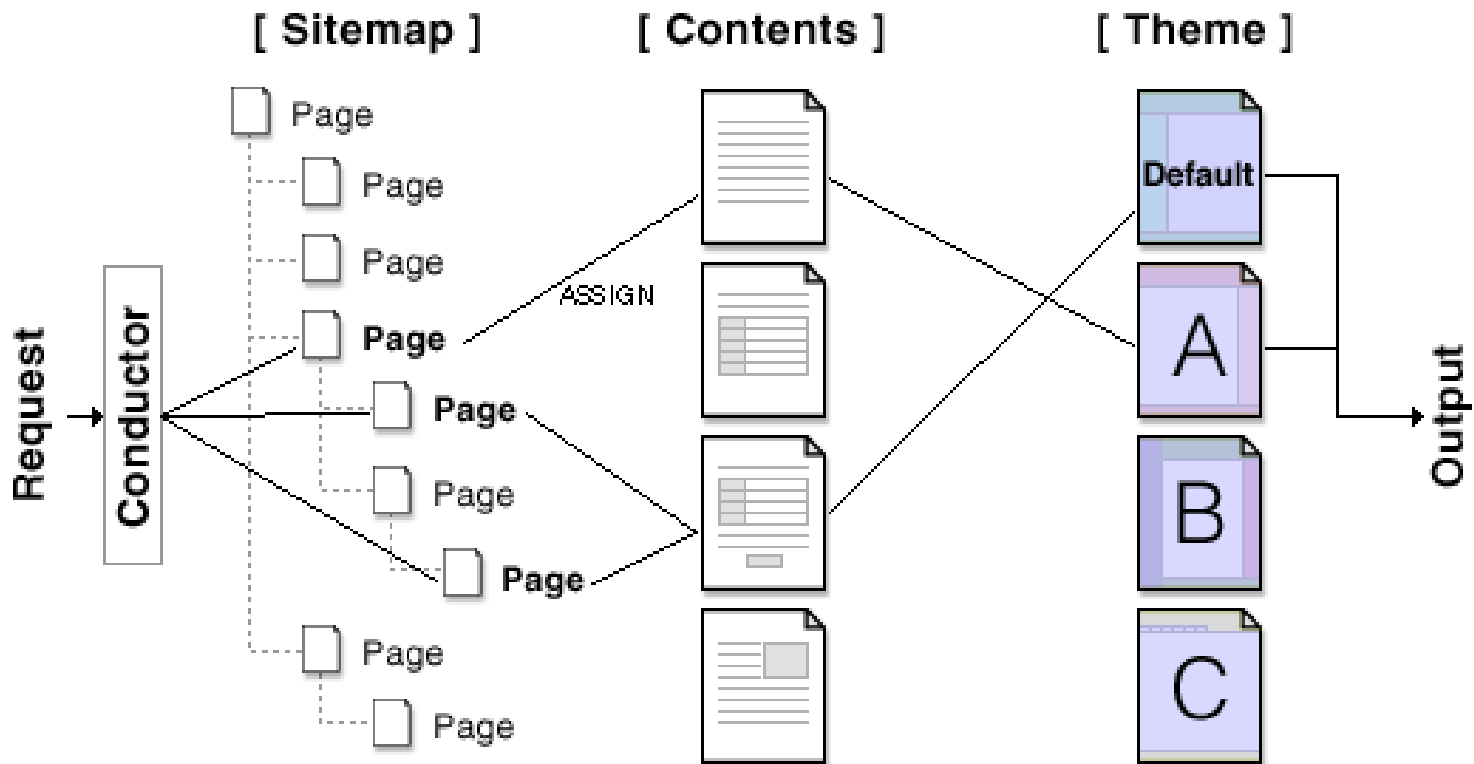
「テーマ」とは

「テーマ」とは、サイト全体のナビゲーション部分や、
トーン＆マナーを一元管理する機構です。



コンテンツ以外の領域は、全てテーマが
一元的に管理しています。
プログラム制御により、ナビゲーションや
パンくずの実装を自動化します。

最後に、Pickles Framework が組み立てて完成！



Pickles Framework は、
これら3種の要素を自動的に合成して、
1つのウェブサイトを構築するフレームワークです。

【仕上げ】 PX=publish で静的ファイルに書き出す

最終的な納品ファイルは、静的なHTMLドキュメント

Pickles Framework は、最終的な制作物を、静的なHTMLの形式に出力する機能を内包しています。クライアントへ納品されるのは、この静的なHTMLです。

【パブリッシュの実行手順】

- 【1】URLの最後に ?PX=publish を付けてアクセスする。
- 【2】パブリッシュボタンを押す。
- 【3】パブリッシュが始まるので、終わるまでしばらく待つ。
- 【4】./_PX/_sys/publish/* に出力される。

特徴

覚えるコトが少ないから、学習コストが低い

■データ構造に、一般的なHTML形式を採用しているので、特殊な技術を習得しなくても、導入できる。

■サイト全体の構造を管理するサイトマップが CSV形式 のため、ノン・エンジニアのディレクターでも直接編集できる。

■カンタンな操作で、インストールからパブリッシュまで行うことができる。

■既定のHTML構造設計の定義を持っているので、サイトごとのコーディング規則などを毎回学習しなおす必要がない(※1)。

※1: Pickles Frameworkの共通ルールで構築されたサイトであることが前提。

**動的サイトとしての運用も。
静的HTMLへのパブリッシュも。**

■最終的に「静的なHTMLとしてパブリッシュしたものを納品する」のが普通の使い方。

■でも、動的なまま公開して、ウェブアプリとして運用することも可能。

初期開発のみ請負い、運用はクライアントが行うような場合にも、静的にパブリッシュした普通のHTMLのみを納品することで、切り分けも容易にできます。

また、万が一、プロジェクトの途中でPxFWがフィットしないと判断したとしても・・・、静的なHTMLを出力し、途中から手作業に切り替える判断も可能。

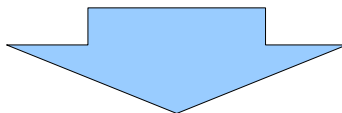
Windowsマシンでも動作します。
つまり、自分のローカル環境でも、
構築することができる、ということ。



XAMPP

XAMPPは、Windows PCに、Apache、PHP、MySQL をカンタンにインストールできるツールです。

<http://www.apachefriends.org/jp/xampp-windows.html>



これを使って、自分のパソコンに、
カンタンに開発環境を構築できます。

※パソコンがMacのひとは、MAMPが使えます。

一般的なCMSとは、どう違う？

■プロのウェブ制作者のために最適化されている。

一般的なCMSは、ウェブリテラシーのない人が編集できるように作られている。

■コンテンツデータを、構造化されたHTMLとして格納する。

一般的なCMSは、データベースに格納してしまう。

■任意の制作環境・テキストエディタで制作できる。

■エンジニアとノン・エンジニアとの共同作業も可能。

SSI(静的なインクルード)と、どう違う？

■ある程度無計画のまま制作を進めても、なんとかなる。

SSIは、事前にしっかりした計画がないと成立しません。(あとから共通領域を足す、など
はできない)

■”ページによって異なる値”にも対応できる。

SSIでは、メタkeywords, description や OGタグ のような、”ページによって異なる値” を
入れる要件には対応できません。

パブリッシュ後の静的なHTML上に、SSIの記述を残すこ
とができるので、SSIとの併用も可能です。

効能 (Pickles Framework 導入で、ウェブ制作はどう変わる?)

機械的な作業は、機械にやらせてもらおうじゃないか！

■「決まったルールに則ってコピーする」などの、機械的な作業は、機械が勝手に行います。

■コーディングにかかる工数が削減されます。

■メタタグ、OGタグなどの「ブラウザから見えない要素」の実装・確認も一覧から行えます。

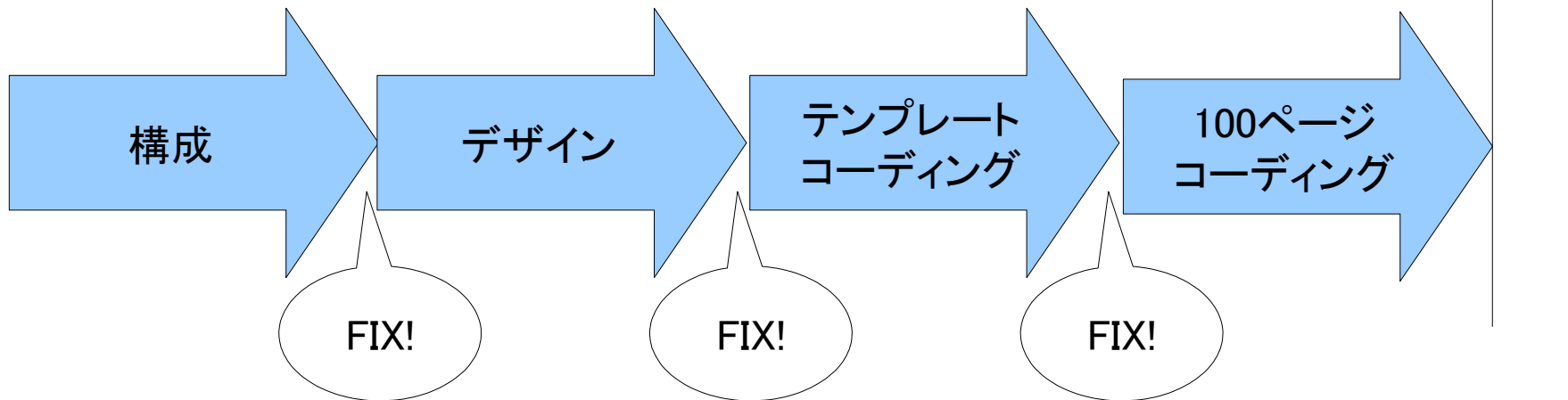
■デフォルトとして設計された汎用的なHTML構造を持っているので、作業手順を固定化しやすくなります。

■ブックマークレットやマクロなど、
制作補助ツールの開発と共通化が容易になります。

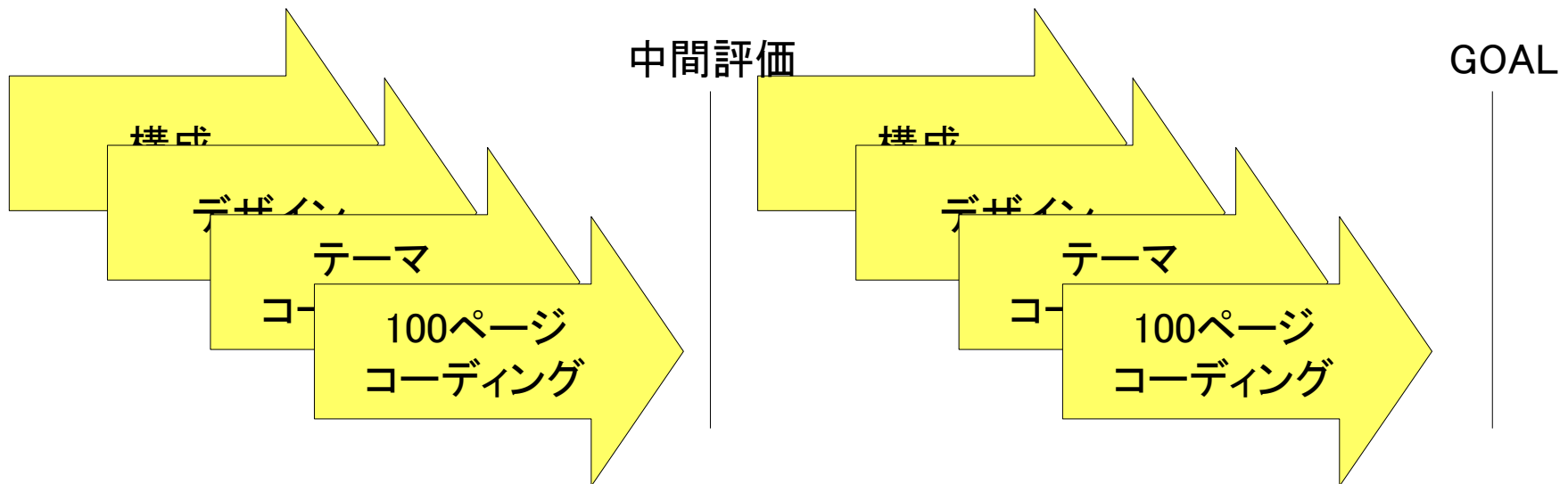
- サイトの全体構造を機械的に生成するので、
「メタタグのコピペミス」などイージー・ミスが劇的に減ります。
- ブックマークレットやマクロの共有化も、
ミスの低減に貢献します。

効能 ～ アジャイル開発フローが可能

■一般的な制作の進め方



■制作プロセスを同時に進行することが可能！

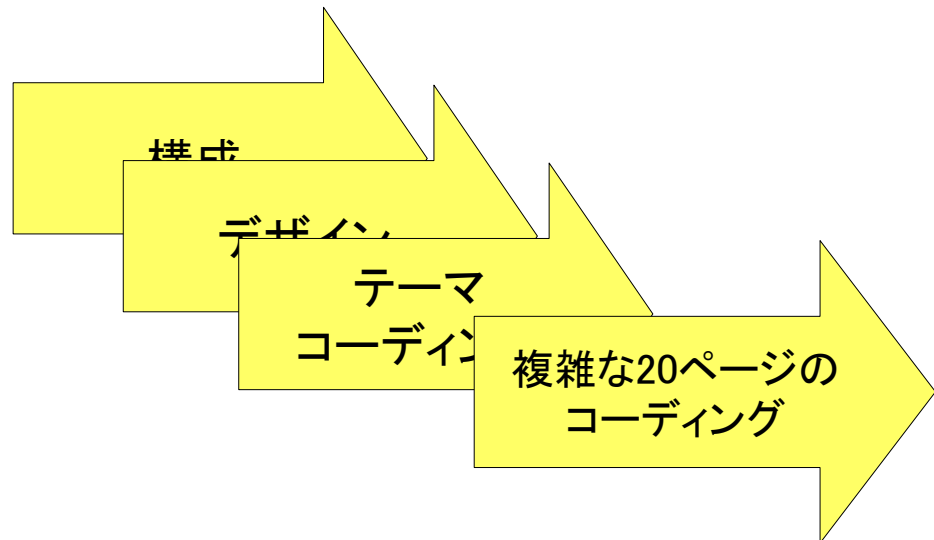
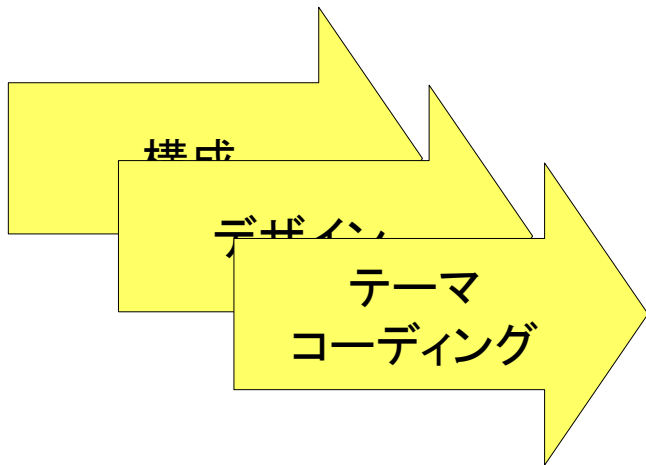


効能 ～ パート別に進め方を変えられる

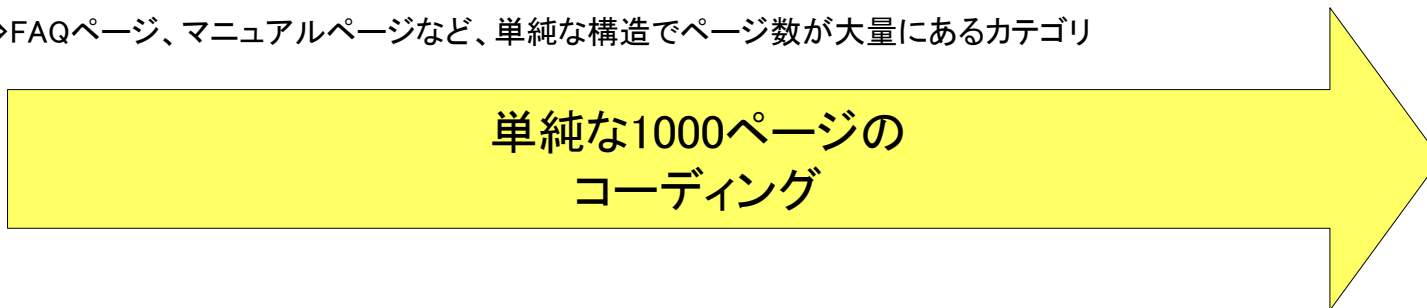
中間評価

GOAL

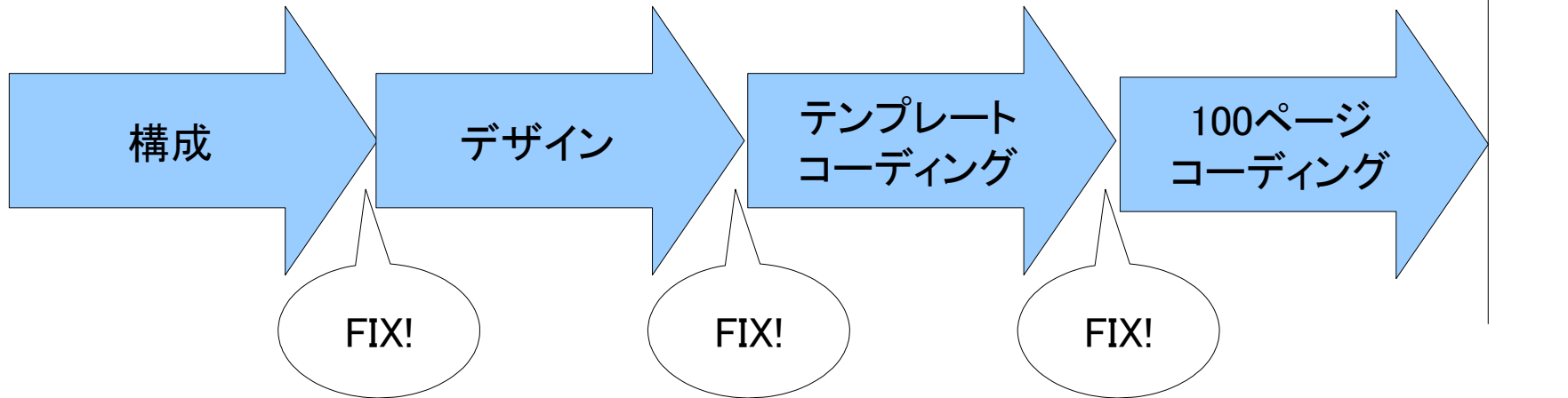
⇒サイト全体設計・構築プロセス



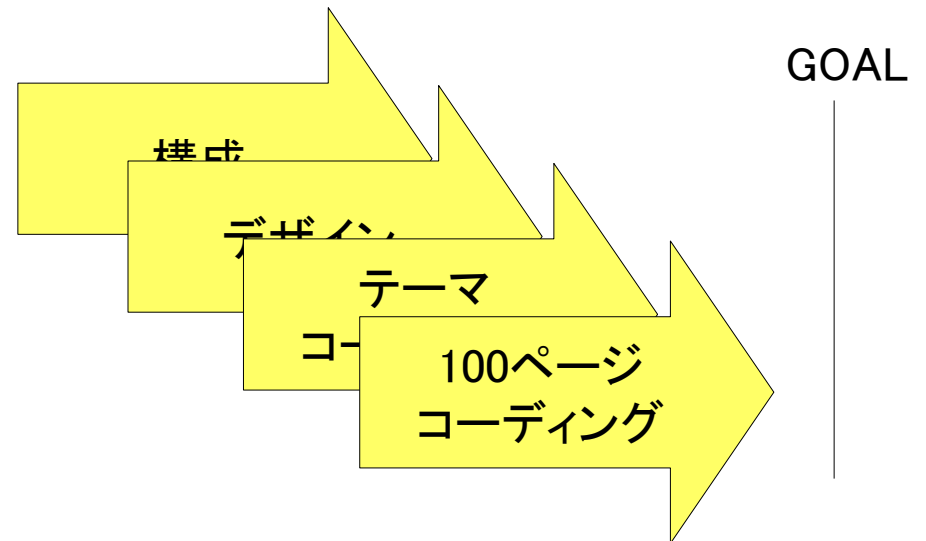
⇒FAQページ、マニュアルページなど、単純な構造でページ数が大量にあるカテゴリ



■一般的な制作の進め方



■制作プロセスを同時に進行することが可能！

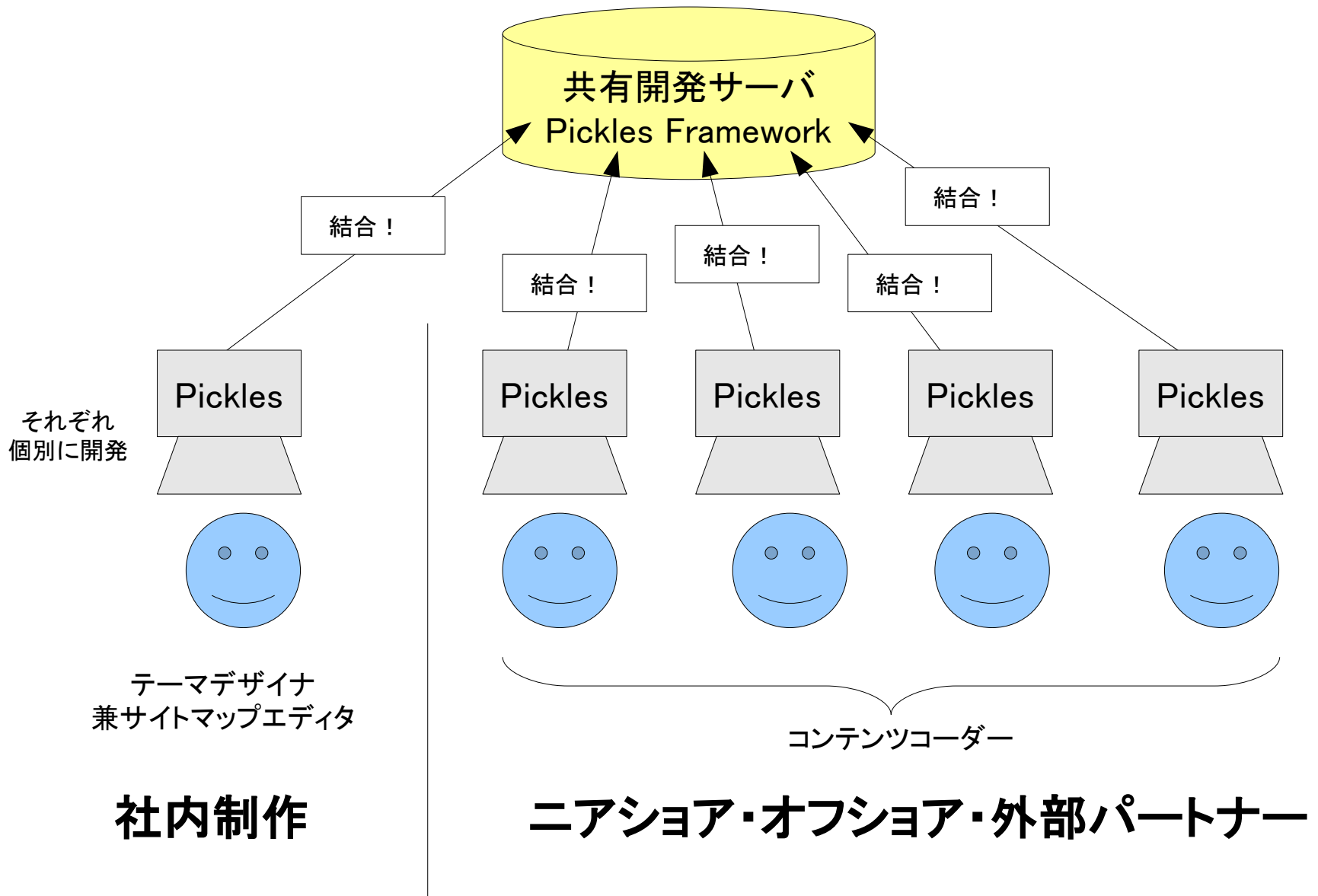


- 時代と共に変化するSNSやSEOの要件に対応できます。
- メタタグやOGタグなどの、ページによって異なる文字列も、CSVから自動的に出力されます。

**■評価フェーズであがったサイト全体に係る問題点について、
改善を実施することができます。**

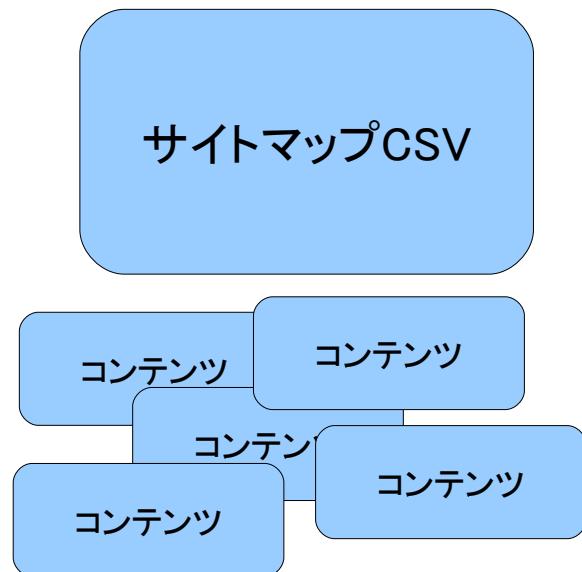
従来のような手作りのウェブでは、いちから作り直しになるほどのインパクトがあり、コストや期間がかかり過ぎるため、改善を実際に実施することは困難です。

効能 ～ 外部のパートナーとの共同制作のリスクを低減(オフショア・ニアショア)

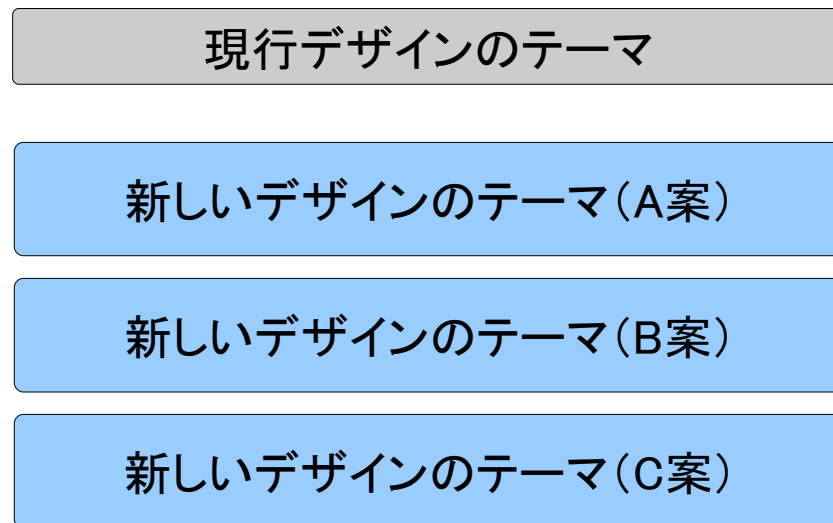


効能 ～ 「リニューアル」と「通常更新」を同時進行

通常運営班の作業対象



デザインリニューアル班の作業対象



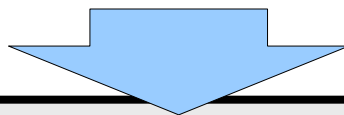
- 作業の対象が物理的に分離されているので、制作を同時進行することができます。
- リアルタイムに最新のコンテンツでリニューアル作業を進行できます。
- 新デザインFIX後のコンテンツ移行作業がほとんど必要ありません。

■構造とスタイルが明確に分離されるので、デバイス別のスタイルの適用が容易。

■サイトの全体構造が動的に生成されているので、将来登場する未知のデバイスにも柔軟に対応できます。

■制作手順が効率化されることで、設計段階でのモックアップ(仮組み)制作のハードルがぐっと下がる。

■クライアント担当者にとって、書類で見るとより実物に触れるほうが、実感として受け取りやすい。



意思決定のブレを起こしにくくする

未開発(※1)

※1: 現在の実装では、モック制作にエンジニアが必要です。

GUI編集機能＋赤入れ機能

- モックアップを、ノン・エンジニアでも操作できる
GUIから制作することができる。
- クライアントは、ウェブ上のインターフェイスで、
HTMLに直接 “赤入れ” を行うことが可能に。印刷物も不要。
- モックアップを、構成書の代替として機能させる。

未開発

GUIコンテンツ編集機能

- 既定の文書構造を編集できるGUI編集画面を搭載予定。
- HTMLの知識がない人(学生、アルバイトなど)でも、コンテンツコーディングを実施可
- ソースの形式が、
一般的な(ただし高度に定型化された) HTML なので、
ノン・エンジニアが制作したコードを、
エンジニアが引き継ぐことが容易。

未開発

既存サイトの自動取り込み機能

- 既存のウェブサイトをクロールし、Pickles Framework の既定の構造に変換するツール。
- 完全な変換はできないが、通常人の手で行う作業の7～8割前後を自動化できる想定。(ただし、変換元サイトの構造による)
- リニューアル案件の制作コストの大幅な削減が期待できる。

未開発

まとめ

特徴

- 学習コストが低い
- 動的にも静的にも利用できる。
- ローカル環境に制作環境を構築できる
- 一般的なCMSとどう違うのか？
 - プロのウェブ制作者のために最適化されている。
 - コンテンツデータを、データベースではなく、構造化されたHTMLファイルとして格納する。
 - 任意の制作環境・テキストエディタで制作できる。
 - エンジニアとノン・エンジニアとの作業分担も可能。
- SSIとどう違うのか？
 - ある程度無計画のまま制作を進めても、なんとかなる。
 - “ページによって異なる値”にも対応できる。

- 1.大量ページコーディングの効率化
- 2.作業手順を固定化できる
- 3.イージー・ミスを低減
- 4.アジャイル開発フローが可能
- 5.パート別に進め方を変えられる
- 6.短納期の制作も可能
- 7.評価フェーズでのサイト全体にかかる問題改善
- 8.SNS、SEOなどの環境変化への対応
- 9.外部のパートナーとの共同制作のリスクを低減(オフショア・ニアショア)
- 10.「リニューアル」と「通常更新」を同時進行
- 11.マルチデバイス対応への障壁をなくす
- 12.設計段階から「モックアップ」を構築
- 13.“構成書”を不要にする
- 14.ノン・エンジニアによる量産コーディング
- 15.大型サイトのリニューアル案件の効率化(半自動化)

ありがとうございました

～Pickles Framework 1.X のダウンロードについて～

Pickles Framework は、オープンソースのフレームワークとして、Github上で有志により開発されています。

下記のサイトから自由にダウンロードして使用してください。

<https://github.com/tomk79/PxFW-1.x>
