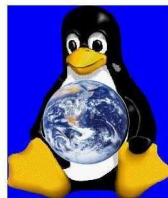


ソフトウェア国際化(I18N)の理念と技術 — 樋浦秀樹氏を偲んで



末廣 陽一
代表取締役
国際化JP株式会社
<http://www.kokusaika.jp/>

樋浦さんへの哀悼

 検索

ホーム ▶ Info ▶ ソフトウェア国際化の巨星落つ、樋浦秀樹氏がUSで死亡

Japanese ▼

ホーム

コンセプト

製品紹介

サービス

OSS

事例紹介

Info

会社概要



ニュースリリース

2010年4月16日
国際化JP株式会社

ソフトウェア国際化の巨星落つ、樋浦秀樹氏がUSで死亡

長年にわたってサンマイクロシステムズの国際化(I18N)アーキテクトを務めた樋浦秀樹氏が4月7日、アメリカで癌のために亡くなった。樋浦氏は、国際化のテクノロジーの発展と普及振興のために多大な貢献を行った。我々は、彼のコンピュータテクノロジーへの造詣の深さ、卓越した技術力、そして情熱と笑顔を忘れることはない。国際化JPIは、心の底から彼の業績とコンピュータ業界への貢献に敬意を表するものである。

友人として そして同志として
末廣 陽一
国際化JP株式会社 代表取締役

(注報より1週間を経過して掲載しました)

自己紹介 末廣陽一の略歴

1962年3月	原産地:鹿児島県
1986年3月	九州大学大学院総合理工学研究科情報システム学専攻修士課程修了。英日間の機械翻訳を研究。
1986年4月	日本デジタルイクイップメント株式会社(日本DEC)に入社。日本DEC研究開発センター株式会社に配属。
1998年10月	会社合併により、コンパックコンピュータ株式会社に編入。
2002年11月	会社合併により、日本ヒューレット・パカード株式会社に編入。
2003年12月	日本ヒューレット・パカード株式会社退社。
2004年2月	ビジネスサーチテクノロジー株式会社を共同で設立。 取締役 研究開発センター ソフトウェア国際化開発部部長に就任。 オープンシステム技術開発部部長を兼任。
2006年4月	ビジネスサーチテクノロジー株式会社退社。
2006年6月	国際化JP株式会社を設立。代表取締役に就任。現在に至る。

自己紹介 国際化についての経歴

- DEC, Compaq, HPでUNIX OSの国際化開発、開発マネージメント、プロダクトマネージメントを担当。日本のTru64 UNIX OS開発チームのプロジェクトリーダーを約10年務める。HP退職前の1-2年は、USやヨーロッパなど世界中の国際化チームの作業を日本に集約。そのマネージメントを務めた。
- UNIX、国際化の標準化委員で会社代表、日本代表団の一員として活動
X/Open 国際化 WG (DEC代表)、 SC22/C Language (JIS、ISO) 委員、
SC22/POSIX (JIS、ISO) 委員
- IBM木戸氏、Sun(当時)樋浦氏らとLinux国際化機能の標準化組織LI18NUX(その後、OpenI18Nに名称変更)を設立。I18N system architecture subgroup のリーダーとして、Linux OSの国際化仕様Li18nux2000をまとめた。
小山教授ともこの時期に出会い、日本のコミュニティメンバーと共に活動した。
- 著書: 共立出版「国際化プログラミング」(エディタとしてまとめた)。Software Design、Computer Todayなどへの記事執筆。
- Compaq/HP時代には社外イベント(Linux World等)や会社主催のシンポジウムなどでソフトウェア国際化についての講演を実施。国際化JP起業後も CICC のイベントなどで国内、国外(タイ、中国、シンガポール、ネパールなど)でソフトウェアの国際化について講演。

ソフトウェアの国際化とは

- **国際化 Internationalization**

「複数の国や地域の文化、言語および習慣を扱うためにコンピュータ・システム/ソフトウェアを汎用化するプロセス」
(共立出版「国際化プログラミング」第1章より)

- **Internationalization** → **I18N**

I と n の間に18 文字ある

- **地域化 Localization**

二つの意味で使われる

- 日本語化, 中国語化, ドイツ語化などの各国語対応を行うこと

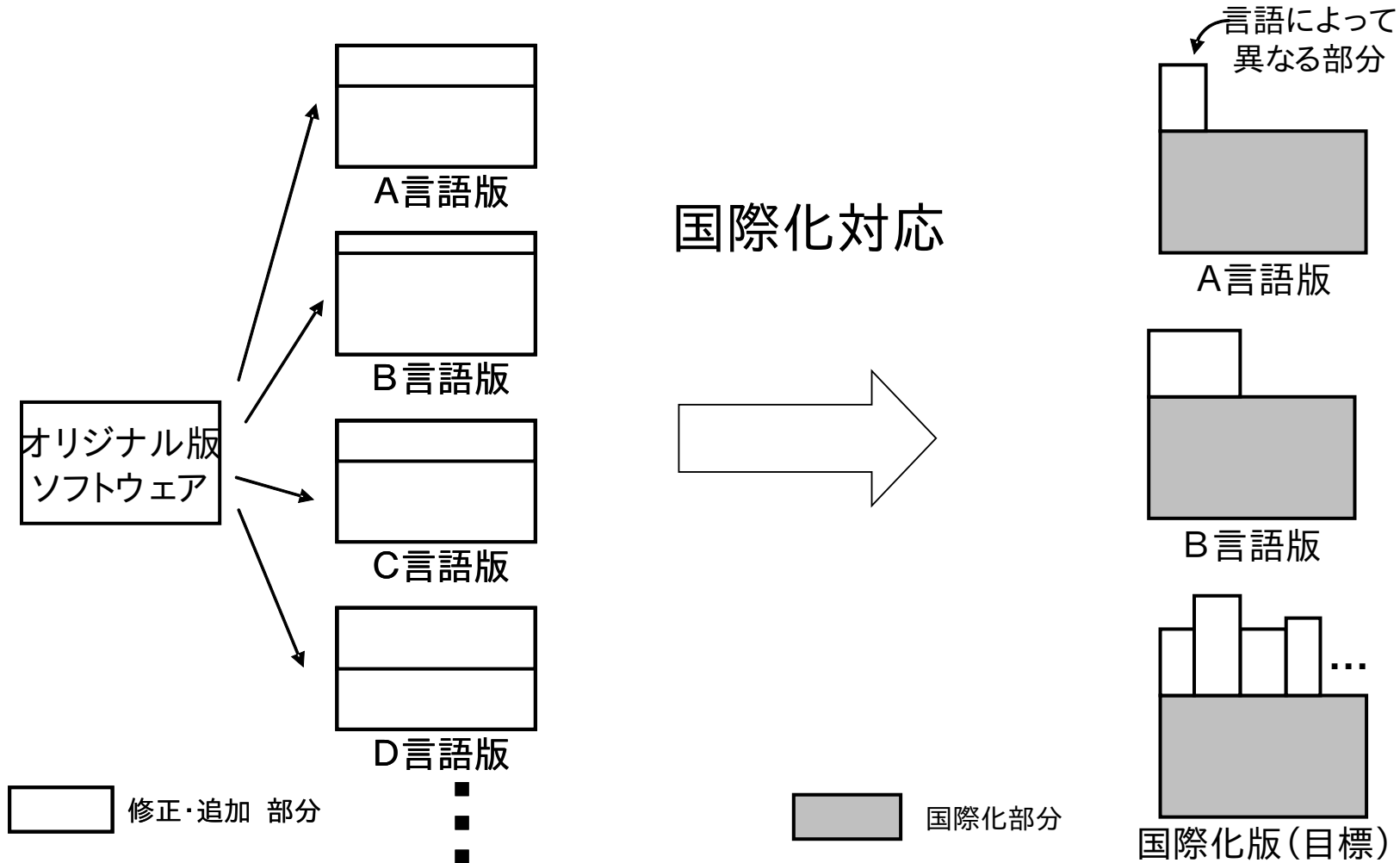
- I18N を実現した後、各国語対応を行うこと

- **L10N は後者 (I18N が前提)**

ソフトウェアの国際化とは

- 国際化≠英語版を作ること
- 国際化≠翻訳
- 国際化は、ほとんどの作業を翻訳ですませることができるようにするための準備作業。
- プログラミングと言語知識とを分離=> それぞれの専門家による効率的な作業を可能にする。

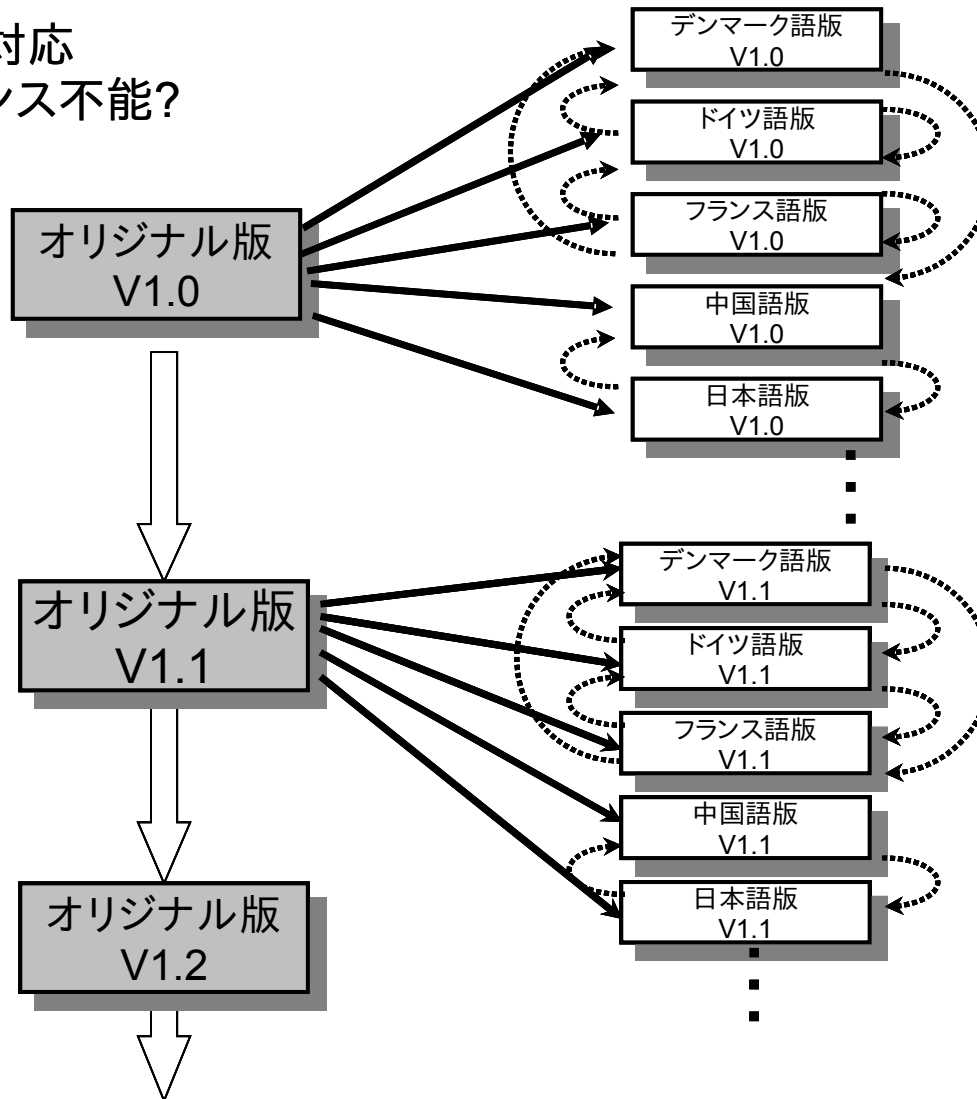
ソフトウェアの国際化とは



国際化を考慮しない開発

複数のバージョン対応
ここまで来るとメンテナンス不能?

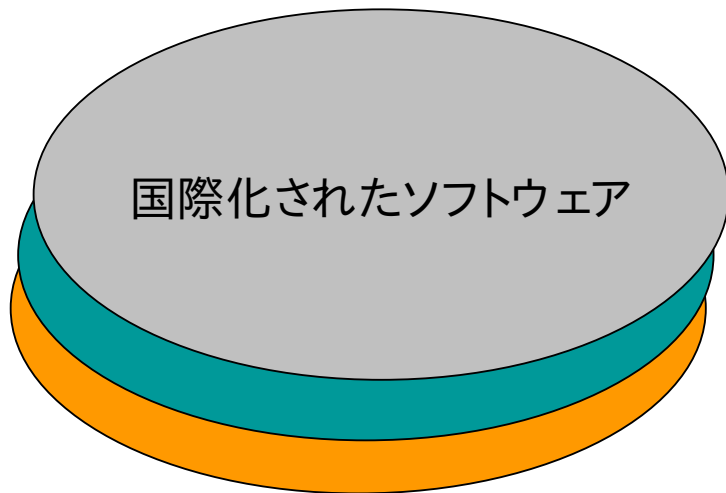
ローカリゼーションは
時間と手間がかかる。
うまく管理しないと、
信じられないくらい大きな
無駄が発生する。



国際化されたソフトウェア

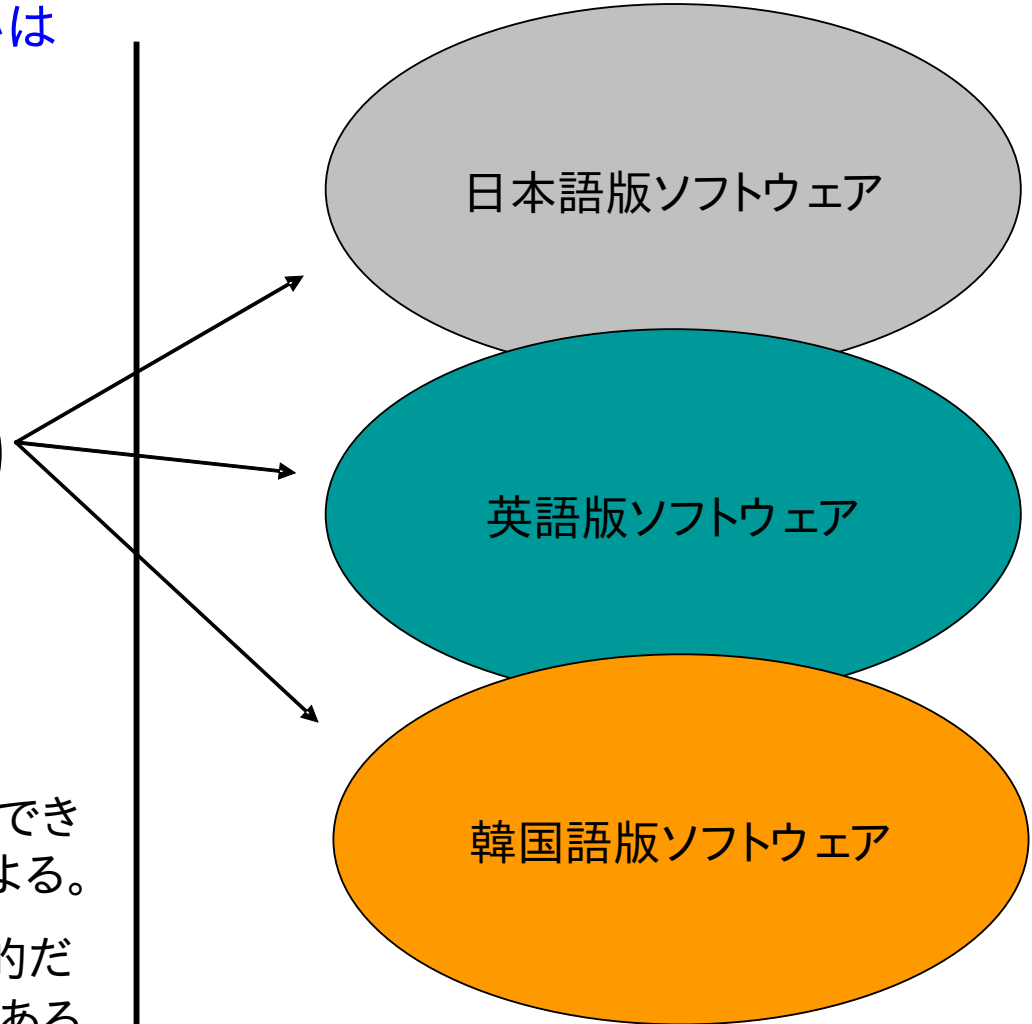
環境ごとに動作が自動で、あるいは
設定で切り替わる

同じソフトウェアで複数の言語を使い
分けることができる。

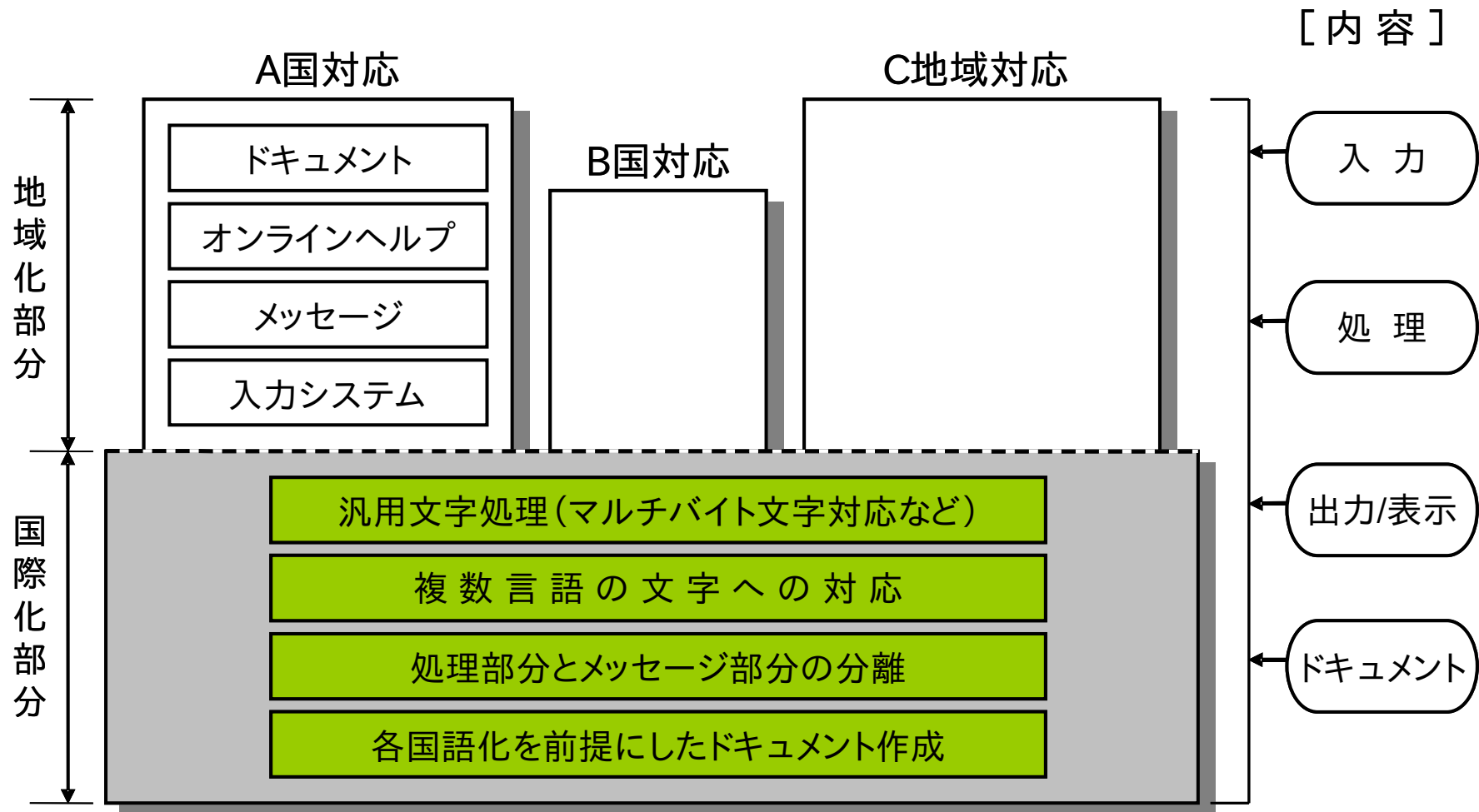


どのような単位で切り替えることができるのかは、ソフトウェアの作り方による。

OSの環境で切り替わるのが一般的だが、アプリが独自に管理することもある。

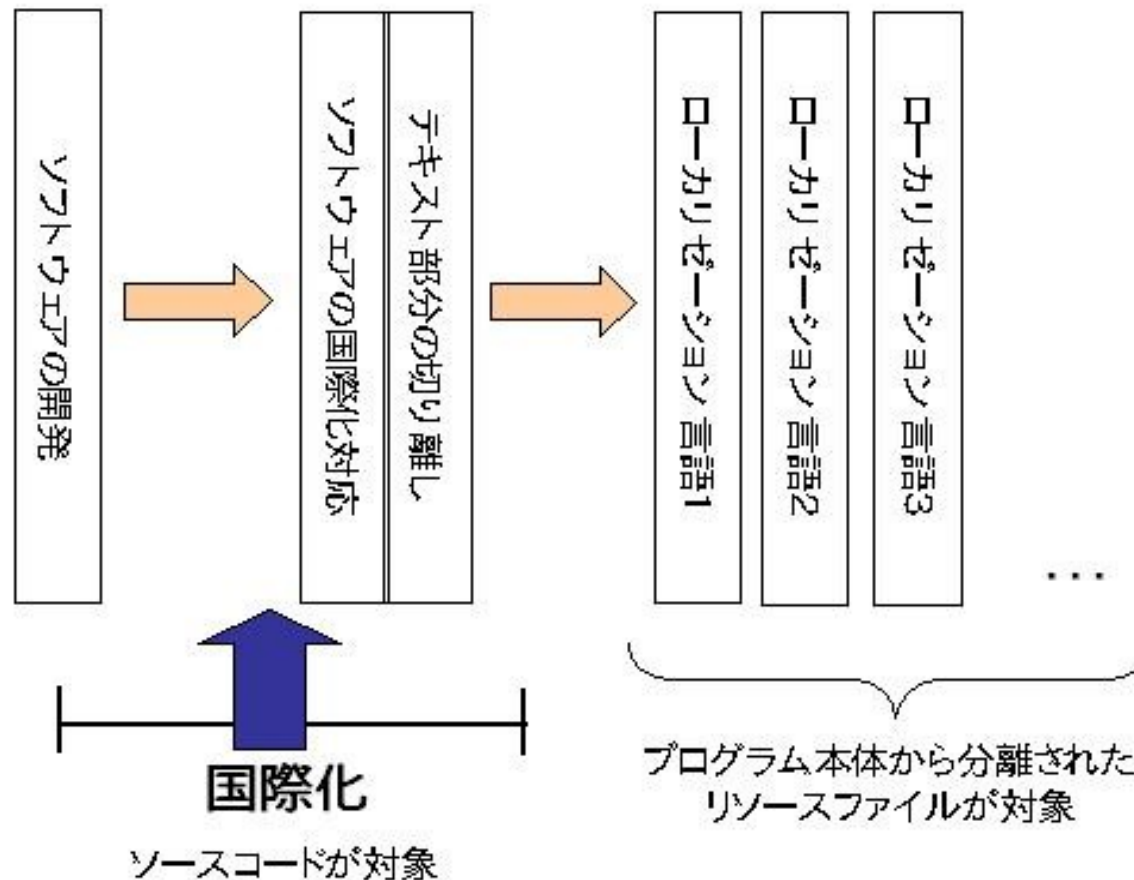


ソフトウェアの国際化とは



ソフトウェア国際化開発のプロセス

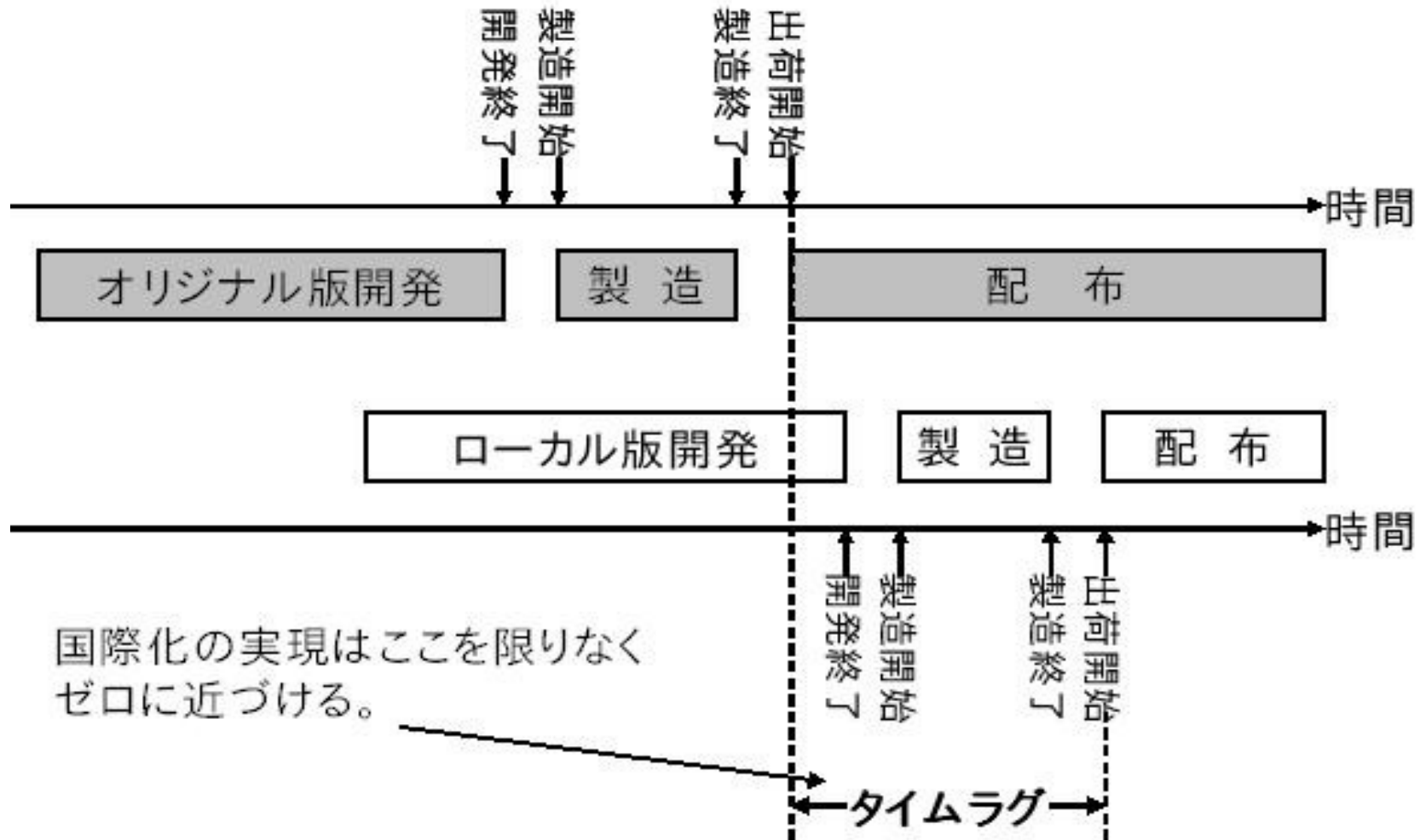
- 開発プロセスの分離 (I18N+L10N)



(狭義の)国際化作業は、

- ソースコードが対象
- 翻訳者ではなく、プログラム開発者が対象
- 社内の開発者が行うことが多い(知財の問題)
- 実務経験を持つ人材は少ない
- 開発プロセスの全体に影響する

ソフトウェアの国際化とは

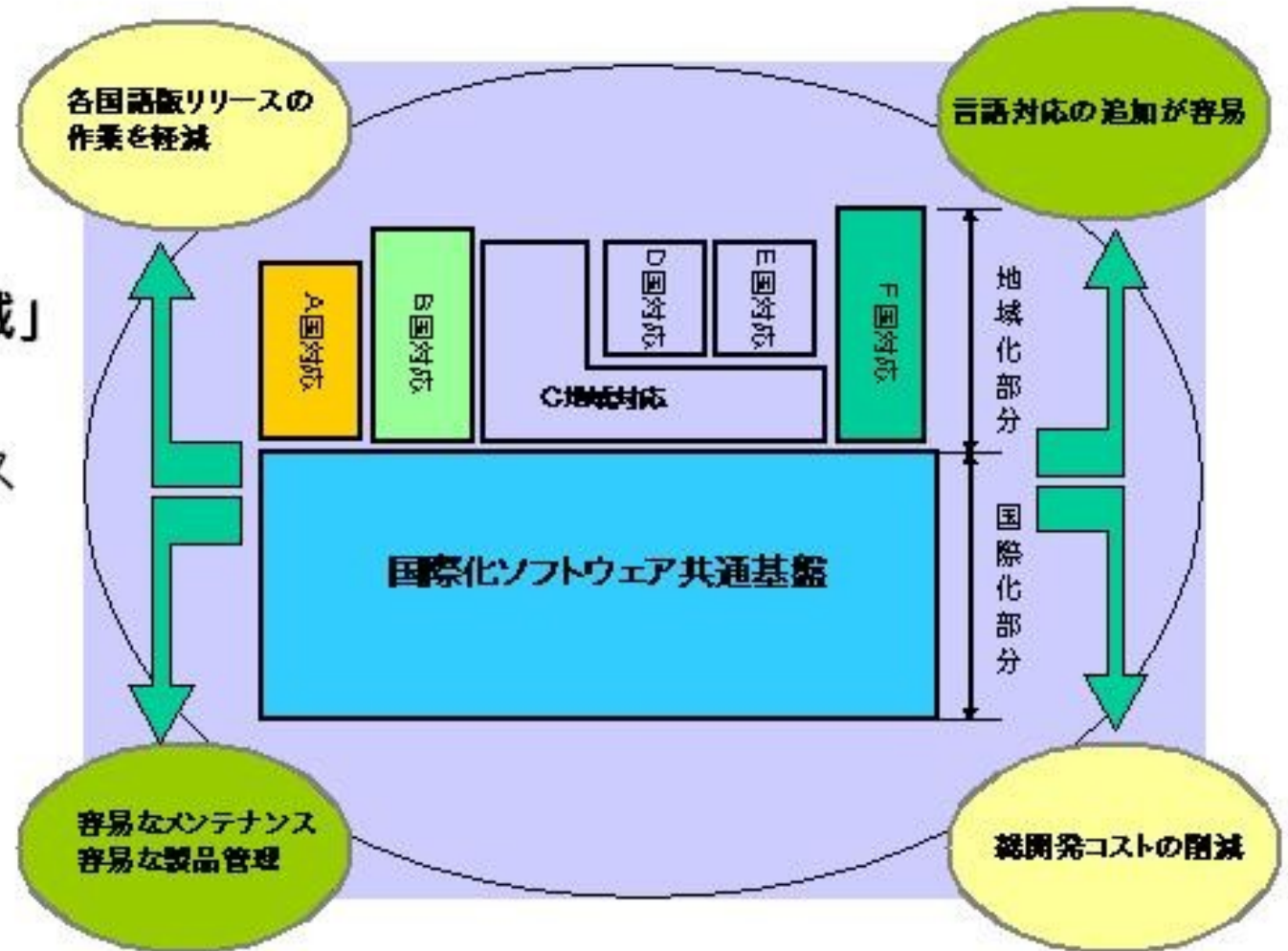


オリジナル版とローカル版(言語対応版)の出荷のタイムラグ

ソフトウェア国際化の利点

- 「開発量の削減」
- 「高品質の確保」
- 「管理の容易性」
- 「トータルコストの削減」

→ グローバルなビジネスの遂行が圧倒的に有利に



ソフトウェア国際化の理念

- 英語も一つの言語
- それぞれの国や地域、言語、習慣を尊重する
- 国際化版の仕組みを作る。各言語への対応は独立して開発することができる
- 今日使われている”グローバルイゼーション”という言葉とは異なるアプローチ
- 標準化。理想と現実。

ソフトウェア国際化の技術

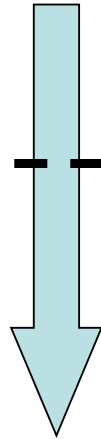
- どうやって国際化を実現するか？
到達点：**シングルソース&シングルバイナリ**の実現
- **シングルソース&シングルバイナリ**を実現するための基礎技術
 - 環境を切り替える仕組み（ロケールなど）
 - 文字や環境依存のデータを汎用的に処理する仕組み（データ型、言語や文化に依存しないAPI）
 - 標準化が重要（ほぼ実現できている。これができないとなんでも自分で抱え持たないといけないことになる。）
 - 依然として、開発者の意識に依存（ここをなんとかしないといけない）

ソフトウェア国際化の技術

動作の仕組み

国際化API

どの言語でも呼び出すAPIは同じ
プログラマは、実行環境を意識する
必要がない



C言語の場合の実装例:
setlocale関数が呼び出された時点で、環境変数の設定に従って、国際化APIの参照先が変更される

複数のAPIを用意して切り替える方法やデータだけを切り替える方法など、実装方法はいろいろある。

下のレイヤーががんばっている
環境に応じた適切な動作をするように作られている

特定言語環境用処理を
切り替え

切替の例:

日付の表現方法

文字種の判定

1文字の取得

ソフトウェア国際化の技術

- 文字コード(Unicode)
 - 知らないところ(内部コード等)で、皆使っている。目に見えるところ(外部コード)としても、多く利用されるようになってきた。
 - 統一するところに大きな意味がある。(内部コード、通信コードなど)
 - 扱えないと問題が出てくる。(すでに出ている)
- 国際化プログラミング
 - プログラミング言語ごとに異なる作法がある。勉強してもらうしかないが、必要としていない人に教えるのはかなり困難。
 - ツールを利用して問題点を検出するなど、大規模になると品質の確保のための対策が必要。
 - テストの強化が必要。動かしてみても、初めてちゃんと国際化できているかどうかを確認できる。
- 開発支援環境
 - テスト、バグ管理、ドキュメント作成、などまだ国際化開発を加速するツール類の拡充が望まれる。

国際化プログラミング

- 比較的新しい技術は国際化対応が容易
 - .NET, Java, Web上の開発フレームワークなど
 - ただし、何も知らずに国際化対応できるわけではない
- プログラミング言語、プラットフォームごとに異なる作法
 - 開発プロジェクトメンバー間で情報を共有する必要がある
 - 開発スタイル、開発プロセスなどを標準化しておくとうい
 - コメントの入れ方、共通ライブラリの使用、共通のツールの使用など
- テストの工数が馬鹿にならない
 - 言語や文字などがロジックに影響を及ぼさないように作った上で、テストの工数が少なくなるように、重点的にテストすべきポイントのツボを押さえておく
 - ときには少々の翻訳の間違い程度には目をつぶる感覚も必要

ソフトウェア国際化の事業化

- これまでの国際化への様々な取り組み
 - 大企業の社員として、会社の活動の一部
 - NPOのメンバーとして、会社／個人のボランティア
- ⇒ 主にインフラ作り。残念ながら認知度は低い。
- これからの国際化
 - 欧米では20年前から国際化に取り組んでいる。日本で国際化(双方向)が必要とされるのはこれからだと考えている
 - ソフトの主導権が作り手から使い手に移ってきている
- ⇒ しなやかに、あらゆる変化に対応できるテクノロジーが必要とされる。クラウド化でさらに国際化の重要性が増す。

会社紹介 国際化JP株式会社

<http://www.kokusaika.jp/>

- ソフトウェア／ウェブの国際化に関わる製品開発およびサービスを提供。「国際化」開発に特化した世界でもユニークな会社。
- 自社開発製品: World Wide Navi (ワールドワイドナビ)
日本では唯一のソフトウェア国際化ツール
無料版も公開
- 多言語Web構築ソリューション: JoomWeb (ジュームウェブ)
日本ではまだ知られていない多機能OSS CMSの Joomla!
(ジュームラ)を使った多言語ウェブサイト構築サービス
- コンサルティング、トレーニング、セミナー
- OSS分野でも活動 (IPA 公募、おきなわOSSプロジェクト)

ソフトウェア国際化の事業化に取り組む理由

- **ライフワークと使命感**:自分／自社の持つ、知識・経験・能力を最大限に社会に役立てる。15年以上にわたって積み重ねてきたものがある。自分にしかできないことがあるはず。
- **技術**:現在の国際化技術はここ10年ほど進歩がない。開発者が意識しないと国際化できないのではダメ。意識しなくても国際化されているのが理想。
- **社会貢献**:企業、特に日本企業の競争力を高める。日本は人材の流動性が低いため、ノウハウが他社に伝わっていない。そもそも、グローバルなソフトウェア開発の経験者がほとんどいない。今後、クラウドのようなネット越しにソフトを使う世界になると、よいものを広く展開したものが勝ち残る。世界との競争はさらに激しくなるし、逆に誰にでもチャンスがある。日本のソフト業界はもっと強くなれる。

これからの日本のソフト業界の 発展に必要なものを考える

- 昔は困っていることを解決すればよかった
 - 日本にとって必要なことがあれば会社を越えて協力した。
(外資系企業の場合)一番大変なのは社内の説得。標準化は共通の武器となった。
 - 日本にいても解決できなくて多くの日本人が海を渡った
 - 国際化は日本人に使いやすい日本語のソフトを流通させる仕組みであるとともに、日本から海外へソフトを売るための仕組みでもあったのだが。
- 今は何も困っていない？ 与えられることに慣れてしまっていないか？
 - 人材、意欲、環境
 - 樋浦さんとは日米で連携して、日本のソフト業界を世界レベルに導くような仕事があった