



XLIFF: Theory and Reality

Lessons Learned by Medtronic in 4 Years of Everyday XLIFF Use

XLIFF Symposium

Limerick, Ireland

September 22, 2010

Global Translation Solutions



XLIFF: Theory and Reality

Lessons Learned by Medtronic in 4 Years of Everyday XLIFF Use

XLIFF Symposium

Limerick, Ireland

September 22, 2010

ぺちやくちや Version

Global Translation Solutions

Linguistic Fun with PechaKucha

- PechaKucha is Japanese slang:

ぺちやくちや

yak-yak, yakity yak (擬音語)

少女たちはボーイフレンドのことを長いことぺちやくちやしゃべっていた | The girls chattered [prattled / gabbed] on and on about their boyfriends.

Credit: definitions taken from Apple's OS X dictionary app

Linguistic Fun with PechaKucha

- PechaKucha is Japanese slang:

ぺちやくちや

yak-yak, yakity yak (擬音語)

少女たちはボーイフレンドのことを長いことぺちやくちやしゃべっていた | The girls chattered [prattled / gabbed] on and on about their boyfriends.

- Close Relative – MechaKucha:

めちやくちや【滅茶苦茶】

めちやくちやな | absurd, unreasonable; (筋道が立たない) incoherent; (向こう見ずな) reckless; (度外れの) excessive

Credit: definitions taken from Apple's OS X dictionary app

Linguistic Fun with PechaKucha

- PechaKucha is Japanese slang:

ぺちやくちや

yak-yak, yakity yak (擬音語)

少女たちはボーイフレンドのことを長いことぺちやくちやしゃべっていた | The girls chattered [prattled / gabbed] on and on about their boyfriends.

- Close Relative – MechaKucha:

めちやくちや【滅茶苦茶】

めちやくちやな | absurd, unreasonable; (筋道が立たない) incoherent; (向こう見ずな) reckless; (度外れの) excessive

- PechaKucha defined as short, concise presentations

irony¹ |'ɪrənē; 'iərnē|

a state of affairs or an event that seems deliberately contrary to what one expects and is often amusing as a result

Credit: definitions taken from Apple's OS X dictionary app

XLIFFs at Medtronic: Background #1

- 2003: Mistakes in found in Japanese submissions
 - Inconsistent source
 - Inconsistent translation
 - Number errors
 - Date errors
- My group tasked with managing translation of:
 - **Regulatory submissions**, technical manuals, etc. etc...

XLIFFs at Medtronic: Background #1

- 2003: Mistakes in found in Japanese submissions
 - Inconsistent source
 - Inconsistent translation
 - Number errors
 - Date errors
- My group tasked with managing translation of:
 - **Regulatory submissions**, technical manuals, etc. etc...

Requirement: Eliminate translation errors in regulatory submissions

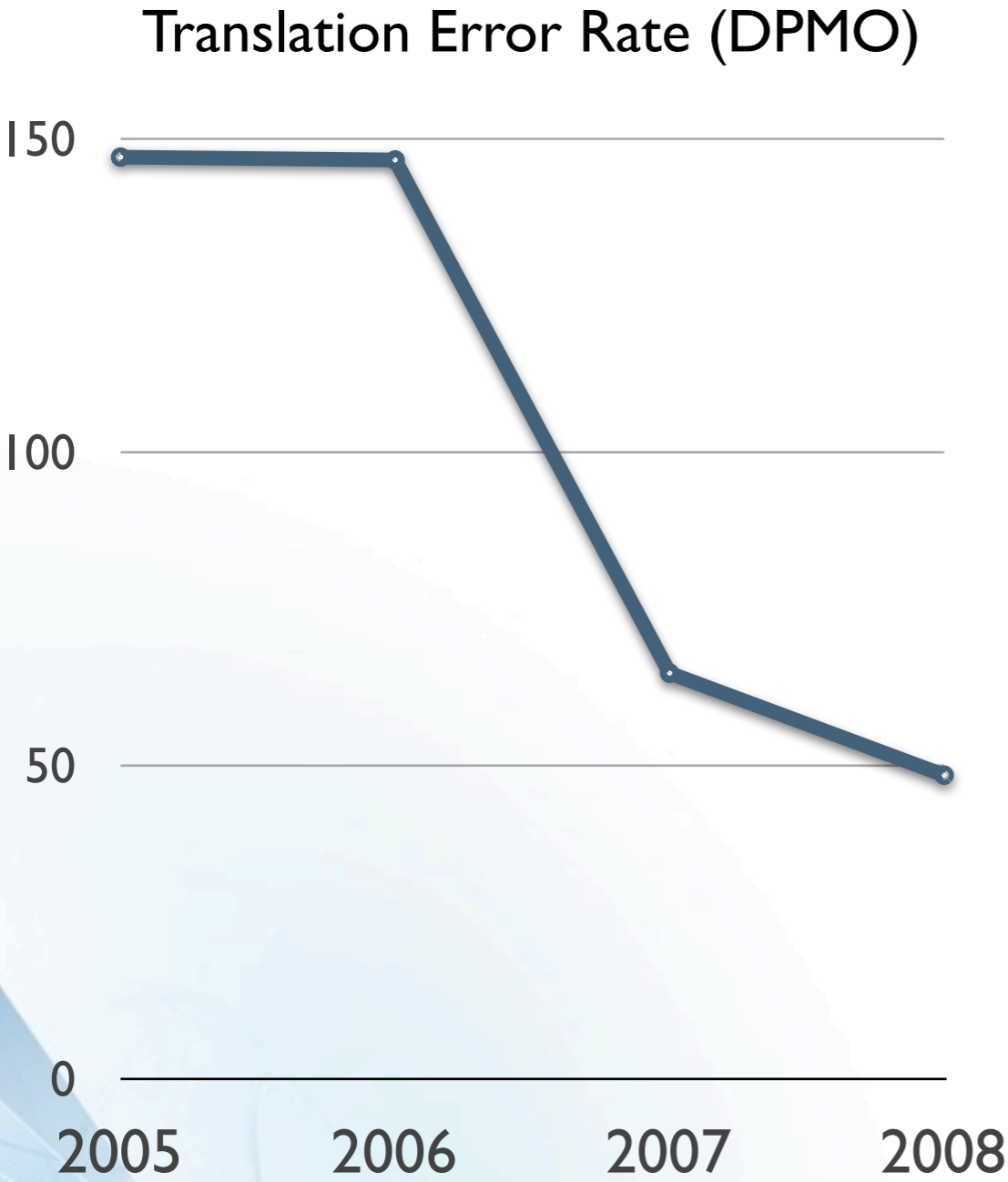
2005 QA Solution - Good and Bad



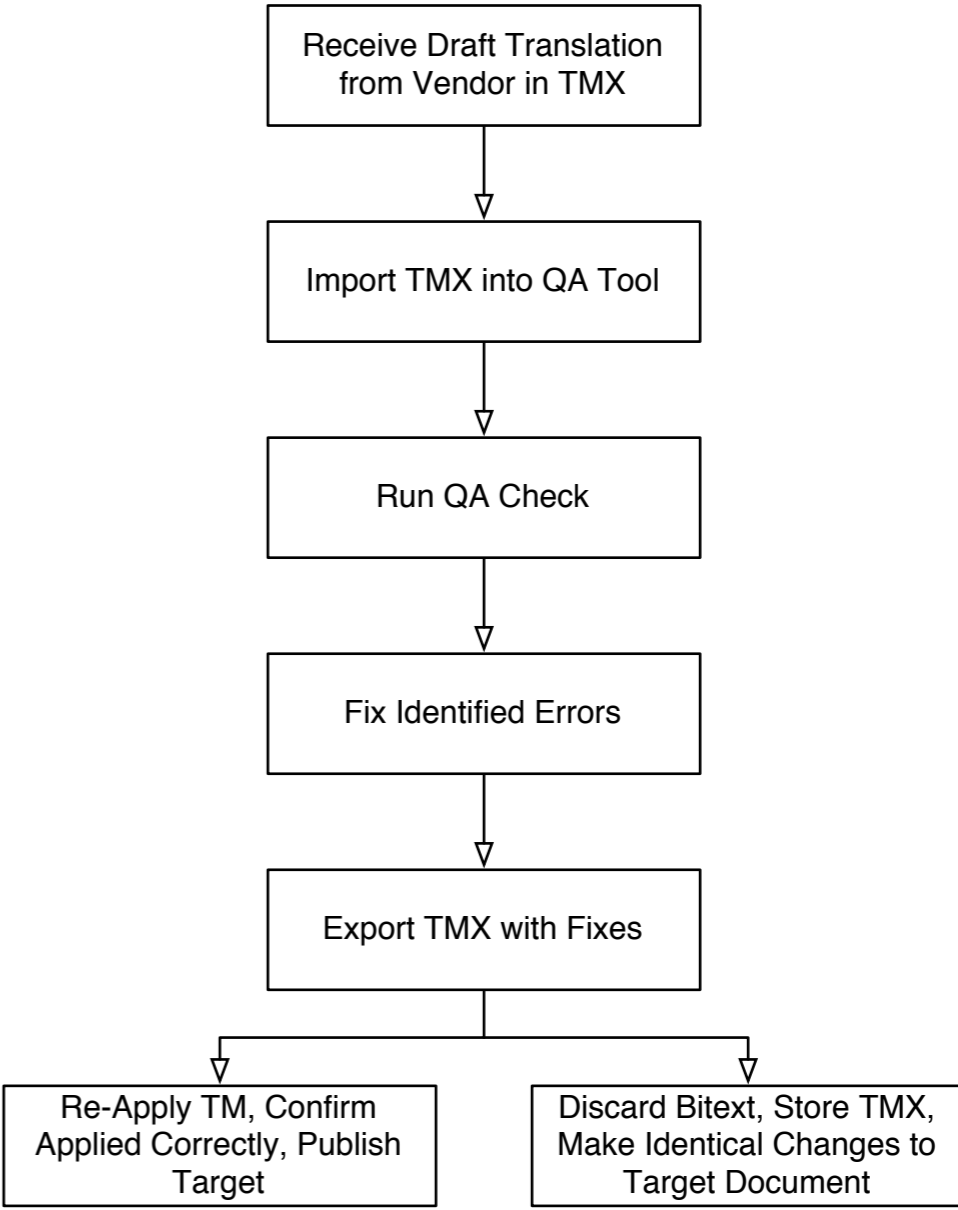
2005 QA Solution - Good and Bad



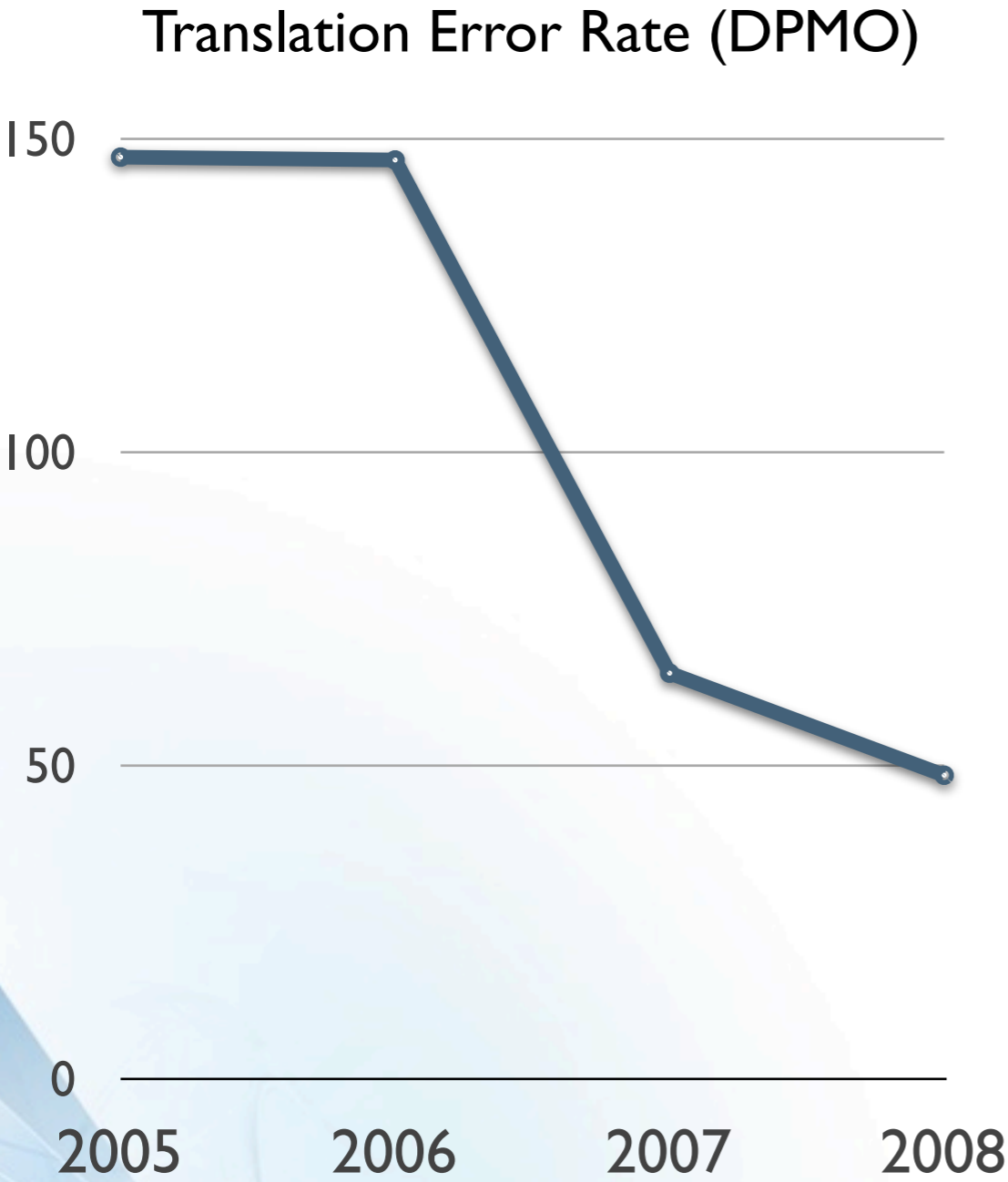
2005 QA Solution - Good and Bad



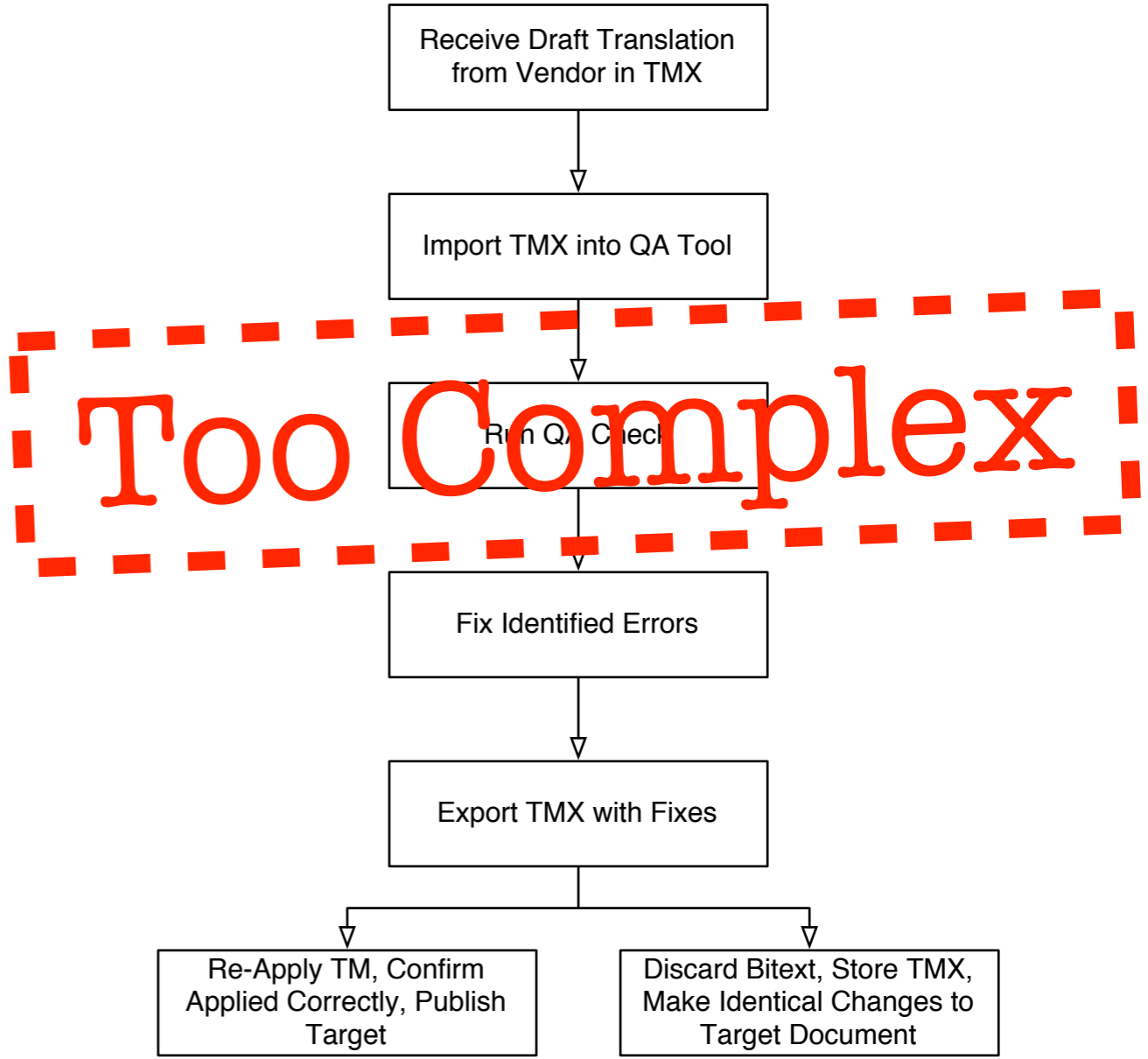
QA Tool Process Flow



2005 QA Solution - Good and Bad



QA Tool Process Flow



XLIFFs at Medtronic: Background #2

- More and more software UI localization
- Switch to Idiom WorldServer
 - Feb 3, 2008: GTS migrates to WS
 - Feb 11, 2008: SDL buys WS, ends development of WS Desktop
- MDT translation group in Europe needs QA tool

Theory of XLIFF

- Tool Interoperability:
 - Any XLIFF can be translated by any XLIFF tool
 - Break lock-in between TM server and translation editor
 - Lossless TM exchange
- Perfect Translation Kit:
 - Meta information
 - TM matches
 - Context information
 - Clipping information (for software strings)
 - Contains everything needed to generate a target

Reality #1: Low Interoperability

- Each creation tool uses the specification differently
- Almost all tools need to use custom extensions
 - Basic, fundamental information being put into extensions
 - Availability of extensions encourages developers to use them even for items that *are* part of the spec
 - Info stored in custom extensions requires reverse engineering
 - Extensions are #1 barrier to interoperability
- Use different tool and...
 - Critical information lost or hidden
 - Work may be lost

Reality #1: Low Interoperability

XLIFF Element	GlobalSight	Heartsome Translation Suite	SDL/Idiom WorldServer	MemoQ 4.2	Okapi Text Extraction	QT	Trados Studio 2009	Sun Open Language Tools Editor	Swordfish XLIFF Editor	XTM	ONTRAM	GlobalLink	MultiTrans
<alt-trans>	●	●	●	○	◐	○	○	●	●	◐	●	●	○
<bin-source>	○	◐	○	○	○	○	○	○	○	○	○	○	○
<bin-target>	○	○	○	○	○	○	○	○	○	○	○	○	○
<bin-unit>	○	○	○	○	○	○	○	○	○	○	○	○	○
<bpt>	●	○	○	●	●	○	○	○	○	○	○	●	●
<bx/>	○	○	○	○	●	○	○	○	○	○	○	●	○
<context-group>	○	○	○	●	○	●	○	○	●	○	○	○	○
<context>	○	○	○	●	○	●	○	○	●	○	○	○	○
<count-group>	○	○	○	○	○	○	○	●	○	○	○	○	○
<count>	○	○	○	○	○	○	○	●	○	○	○	○	○
<ept>	●	○	○	●	●	○	○	○	○	○	○	●	●
<ex/>	○	○	○	○	●	○	○	○	○	○	○	●	○
<external-file>	○	○	○	○	●	○	○	●	●	○	○	●	○
<g>	○	○	○	○	●	○	◐	○	○	●	●	○	○
<glossary>	○	○	○	○	○	○	○	○	○	○	○	○	○
<group>	○	○	○	○	●	●	◐	○	○	◐	●	●	◐
<internal-file>	○	○	○	◐	○	○	◐	○	●	●	○	●	◐
<it>	●	○	○	○	●	○	○	○	○	○	○	●	○
<mrk>	○	○	○	○	●	○	●	○	○	○	○	●	○
<note>	●	●	●	●	●	◐	○	●	●	◐	●	●	○
<ph>	●	●	●	○	●	○	○	○	●	○	●	●	●
<phase-group>	●	○	○	○	○	○	○	○	○	○	○	○	○
<phase>	●	○	○	○	○	○	○	○	○	○	○	○	○
<prop-group>	○	○	○	○	○	○	●	●	○	○	○	○	○
<prop>	○	○	○	○	●	○	●	●	○	○	○	○	○
<reference>	○	○	○	○	○	○	◐	○	○	○	○	○	○
<seg-source>	○	○	○	○	●	○	●	○	○	○	○	○	○
<skl>	○	○	○	●	●	○	○	○	●	●	○	●	●
<source>	●	●	●	●	●	●	●	●	●	●	●	●	●
<sub>	○	○	○	○	○	○	○	○	○	○	○	◐	○
<target>	●	●	●	●	●	●	●	●	●	●	●	●	●
<tool>	○	○	○	●	○	○	○	○	●	●	○	●	○
<x/>	○	○	○	○	●	○	◐	○	○	●	●	●	○

even

Reality #2: Specification Too Broad

- Tool developers can't adequately support all features
- Not enough definition around using critical elements
 - Tool A: uses <x> etc. to show inline tagging of TM match
 - Tool B: uses <x> etc. to show diff between match and TU source
- Too many ways to accomplish same thing:
 - Inline tags: <ph>, <it>, <x>, <g>, <bpt>, <ept>, <bx>, <ex>
 - Source file skeleton:
 - embedded in same file (raw)
 - embedded in same file (compressed/encrypted/etc.)
 - link to separate file
 - link to separate file on server
 - in practice: only the creator tool can generate a target file from an .xlf
 - Translation state: “state” vs “state-qualifier”
 - etc etc

Reality #3: Specification Too Narrow

- Specification completely ignores some critical data
- Basic required information:
 - who last edited a TU or note? when?
 - where do you store the meta info after accepting an <alt-trans> or live TM match?
 - proper behavior when tools support different TU state values?
 - where do you store project info? (title, project id, instructions, etc.)
 - where do you specify which language code system you are using? (for example: ISO 639-1 vs ISO 639-2)
 - where and how do you store QA info?
- Terminology
 - what is the convention for including a TBX or other term file?
 - XLIFF markup to link words in source and/or target to TB?
- Software Context
 - Not enough information to produce facsimile of a UI window
 - How can one store an image of a UI resource so it's available to translators?

XLIFF Reality - Summary

- Not all XLIFF are created equal
- Tool vendors can be compliant with standard, but still lock in users
- Full interoperability is a dream, not a reality

Why Interoperability Matters

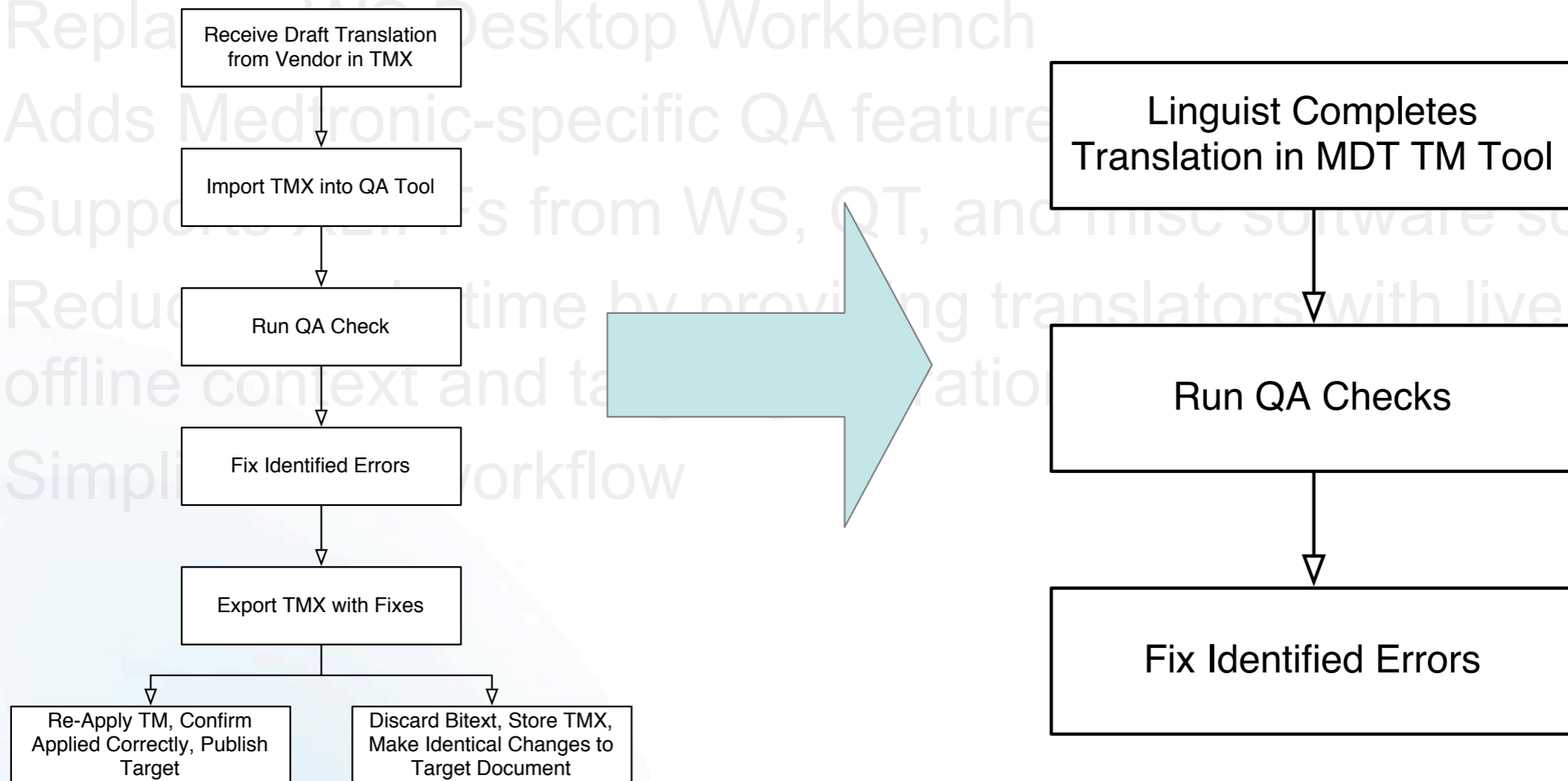
- Single standard makes targeted tools more maintainable
 - Develop for the specification, not for the tool – less tool lock-in
 - Public specification means easier development
 - No longer wholly dependent on tool vendor “professional services”
 - Gain full control of your translation workflow
- Single standard helps commercial tool development
 - Translation tools “industry” not particularly healthy
 - Fragmentation means lower volume for each tool = higher cost
- Allows everyone to use the right tool for the job
 - Freelancers select translation editors based on feature comparison, not compatibility with TM system that created project
 - Enterprise & language vendors free to select TM server system without worrying about acceptance by freelancers
- Bottom Line: Interoperability Saves Consumers Money

XLIFFs at Medtronic - Translation Editor

- Relatively simple to modify TM editor to be XLIFF editor
- Replaces WS Desktop Workbench
- Adds Medtronic-specific QA features and extensions
- Supports XLIFFs from WS, QT, and misc software sources
- Reduces cycle time by providing translators with live, offline context and target generation abilities
- Simplifies QA workflow

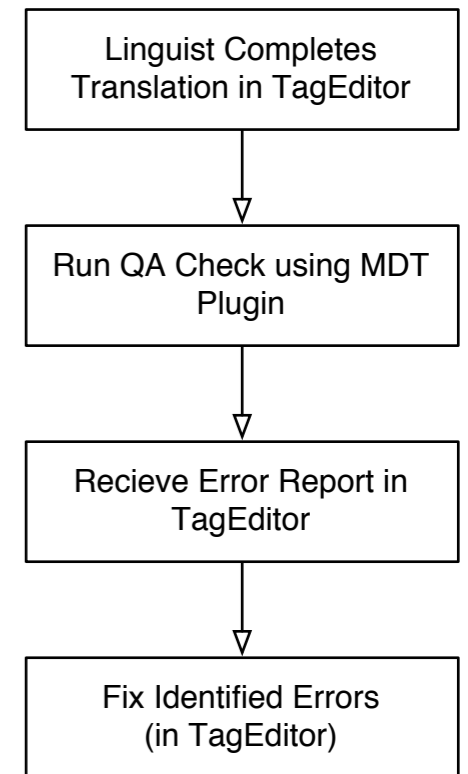
XLIFFs at Medtronic - Translation Editor

- Relatively simple to modify TM editor to be XLIFF editor
- Replace Desktop Workbench
- Adds Medtronic-specific QA features
- Support for files from WS, QT, and misc software sources
- Reduce time by providing translators with live offline context and target information
- Simplify workflow



XLIFFs at Medtronic - QA Plugin

- Minneapolis translation group using XLIFF-based QA Tool
- European translation group using TagEditor (.ttx)
- Solution: TagEditor plugin converts .ttx to .xlf
- Leverages 100% of work done on XLIFF-based QA tool
- Extensible to any workflow that supports .xlf
- *Support for one standard file format makes all parts of translation ecosphere more efficient*



Why XLIFF Matters

- XLIFF is the best (only) hope for interoperability
- Even with its warts, XLIFF is better than completely proprietary formats (.itd, .ttx, etc.)
 - At minimum, basic text data is available to any tool
 - Allows custom tools to access translation files as part of workflow
 - Momentum is forcing tools developers to provide at least basic support
- XLIFF has chance to realize it's full potential