



WORLD
EVENT™

Evangelizing Automated Publishing In Your Organization

Elizabeth Fraley

Single-Sourcing Solutions, Inc.



Prepare To Evangelize

- Create the cause
 - Love the cause
 - Boil it down
 - Be visual
- Do not rush directly into the topic
 - Localize the pain
 - Identify a safe first step
- Learn to give a good demo or presentation
 - Bring in the experts
 - Anticipate the questions people will ask

Evangelize Everywhere

- Build relationships
 - Ignore pedigree
 - Look for agnostics, ignore atheists (Linux vs Windows)
- Communicate
 - Execute (over and over again)
 - Pay attention to reactions
 - Talk straight
 - Demonstrate respect
 - Create transparency

Three Facets Of Documentation Systems

Storage

Maintenance

Creation

Desktop Publishing Systems

20-year old response to custom software

Storage	Binary <ul style="list-style-type: none">• Requires specific vendor application• Not extensible, accessible• No lifecycle control
Maintenance	Time-Consuming <ul style="list-style-type: none">• Repeated information• Repeated styling• Intensive manual effort
Creation	Proprietary <ul style="list-style-type: none">• Creation dictates storage and maintenance choices

Rise of HTML and XML: Succeeding Where SGML Did Not

Responding to the problems with DTP

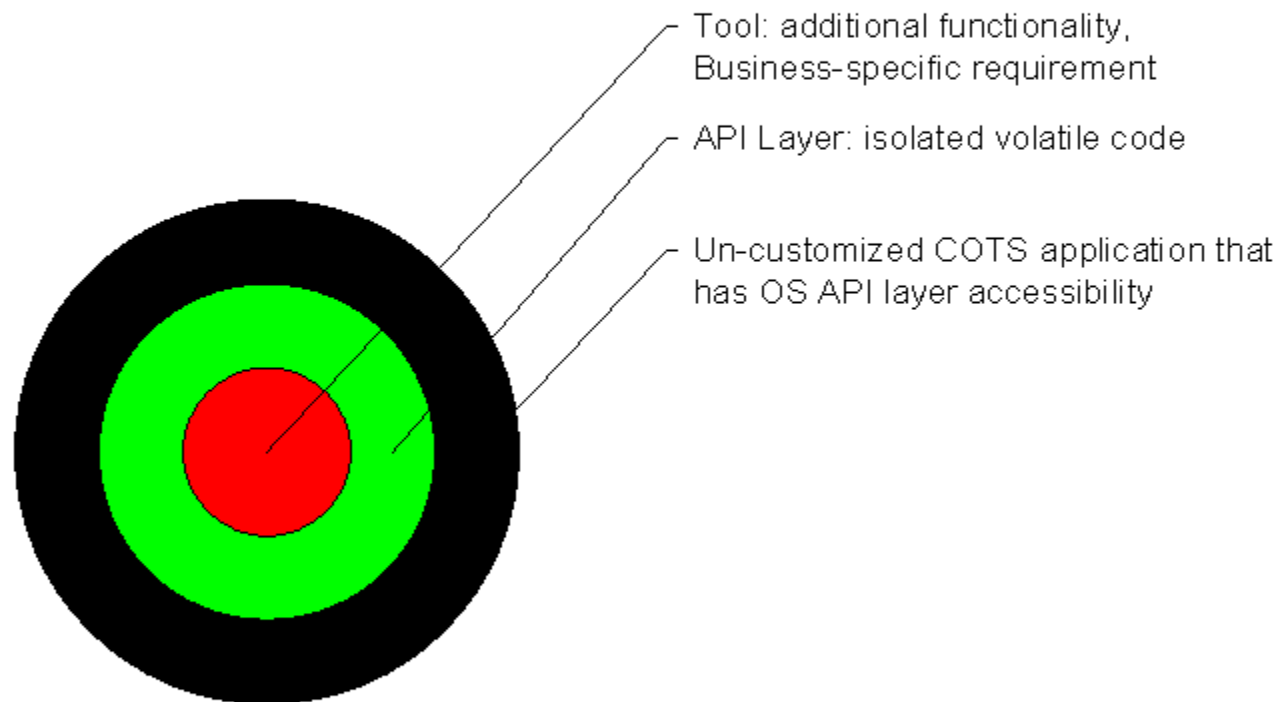
Storage	ASCII <ul style="list-style-type: none">•Human readable•Always available
Maintenance	Libraries <ul style="list-style-type: none">•Common information sets•Reuse information•Automate styling
Creation	Metadata <ul style="list-style-type: none">•Document contains information describing what it contains

Rise of HTML and XML: Succeeding Where SGML Did Not

Responding to the problems with DTP

Storage	XML (DTDs) <ul style="list-style-type: none">•Standards !•MILSPEC/FOSI (SGML)•DocBook•DITA•S1000D
Maintenance	Tool Development <ul style="list-style-type: none">•API layers•Linking Mechanisms (XML technology growing)

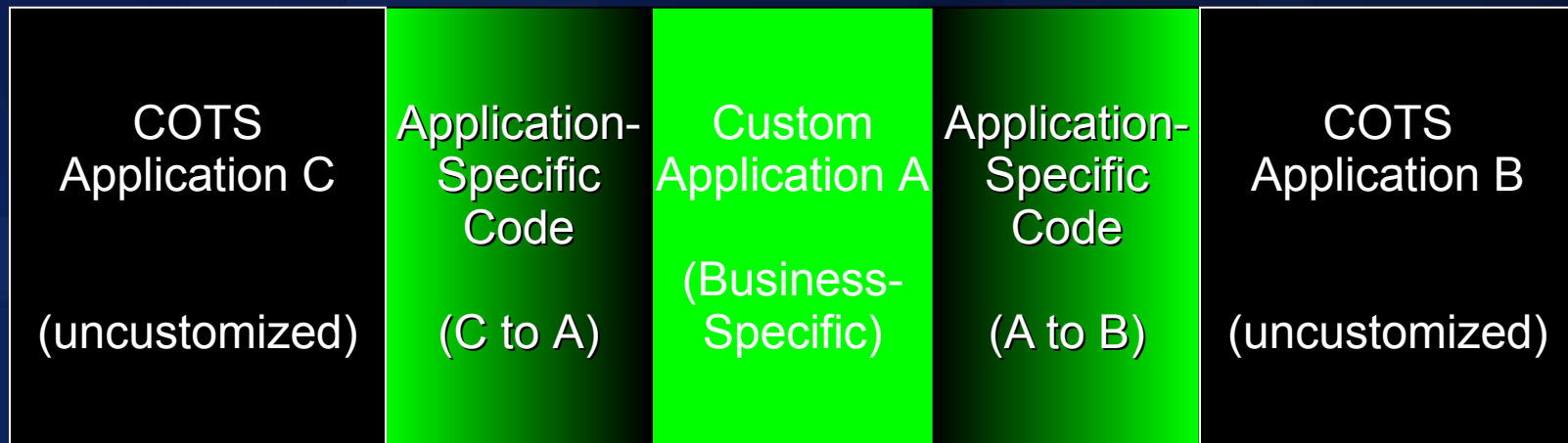
How API Layers and Object Oriented Programming Works



Responsible Software Design:

Use API layer to isolate application specific code that connects any particular tool with any particular application

Software Tool Development: Bridging Applications



Responsible Software Design:

API layers isolate volatile code tied to business-specific requirements and version-specific or application-specific implementations

Rise of HTML and XML: Succeeding Where SGML Did Not

Responding to the problems with DTP

Storage	XML (DTDs) <ul style="list-style-type: none">•Standards !•MILSPEC/FOSI (SGML)•DocBook•DITA•S1000D
Maintenance	Tool Development <ul style="list-style-type: none">•API layers•Linking Mechanisms (XML technology growing)
Creation	CMS/CSDB <ul style="list-style-type: none">•Define “behaviour”•Open source architectures•Search, reporting features

Changes To Documentation Systems

Over the last 20 years

<p>Binary Storage</p> <ul style="list-style-type: none"> • Proprietary – requires specific vendor application • Not extensible, accessible • No lifecycle control 	<p>ASCII</p> <ul style="list-style-type: none"> • Human readable • Always available 	<p>XML (DTDs)</p> <ul style="list-style-type: none"> • Standards ! • MILSPEC/FOSI (SGML) • DocBook • DITA • S1000D
<p>Maintenance</p> <ul style="list-style-type: none"> • Repeated information • Repeated styling • Intensive manual effort 	<p>Libraries</p> <ul style="list-style-type: none"> • Common information sets • Reuse information • Automate styling 	<p>Tool Development</p> <ul style="list-style-type: none"> • API layers • Linking Mechanisms (XML technology growing)
<p>Creation</p> <ul style="list-style-type: none"> • Proprietary • Creation dictates storage and maintenance choices 	<p>Metadata</p> <ul style="list-style-type: none"> • Document contains information describing what it contains 	<p>CMS/CSDB</p> <ul style="list-style-type: none"> • Define “behaviour” • Open source architectures • Search, reporting features

Questions?



Elizabeth Fraley

Single-Sourcing Solutions, Inc.

liz.fraley@single-sourcing.com

408-660-3219