Content Reuse: Great Expectations

Lelia Wright

September 13, 2016

BROCADE

Industry leader in hardware and software-based networking infrastructure solutions, Brocade provides scalable server virtualization and cloud strategies

Founded in 1995, HQ in San Jose, CA

Over 4,550 employees in 36 countries

Customers in 160 countries

$2+ billion in annual revenue

NASDAQ: BRCD

Technical Communications

- ~60 member distributed team
- Hosted SDL Knowledge Center 2014 with XMetaL 9 and World Server
- ~80,000 objects in the CCMS
- ~2,900 publications covering 67 products in 9 product areas
- 800-1,000 publications per year

The Great Expectations Request

We have a new platform, we need a new documentation set, but we have no budget for a new writing team.

You can reuse, can't you?

The Reply:

Yes, we can!

The Reaction:

Help! What did we just promise?

Expectations to Achieve and Manage

Expectation 1:

- Content Reuse solves all your problems so you don't need more writers to create an endless number of new platform documentation sets

Expectation 2:

- All content is written once and all content is reused as is

Expectation 3:

- I don't want to see any other platforms mentioned in my documentation

Expectation 4:

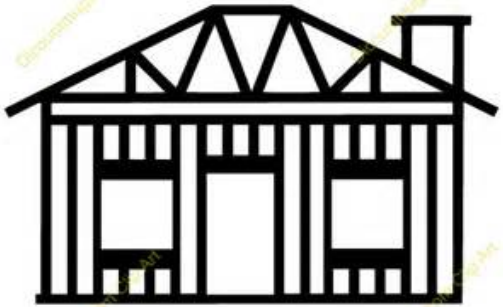
- We need higher quality documentation

The Journey Begins…

Let's cover the basics of how we achieved content reuse and along the way discover how we managed expectations.

Interesting Fact:
Charles Dickens wrote *Great Expectations* in carefully structured weekly installments.

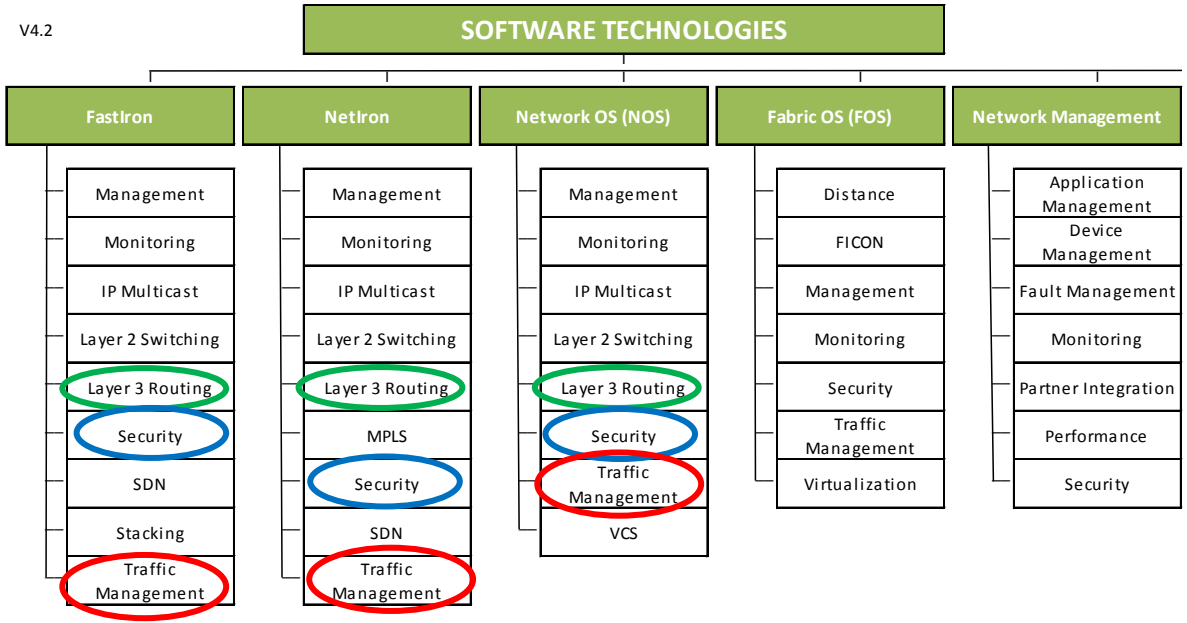
You need a framework



- **Determine how to structure your content – build a taxonomy**
- **Analyze content mapping for reuse possibilities**
- **Use a taxonomy tool or just a spreadsheet**

Our framework

Within our corporate taxonomy



We mapped our content to software technology groups by platforms

Determined our unit of reuse = technology

Match your deliverables to your framework

Layer 3 Configuration Guide

Security Configuration Guide

Traffic Management Configuration Guide

- **These configuration guide titles match our technology groups**
- **Some deliverables stand alone – Licensing guide, for example**

Before:

Product-centric content developers working in silos

Platform A writer

Writes QoS, BGP, RADIUS

Platform A books

Platform B writer

Writes QoS, BGP, RADIUS

Platform B books

Platform C writer

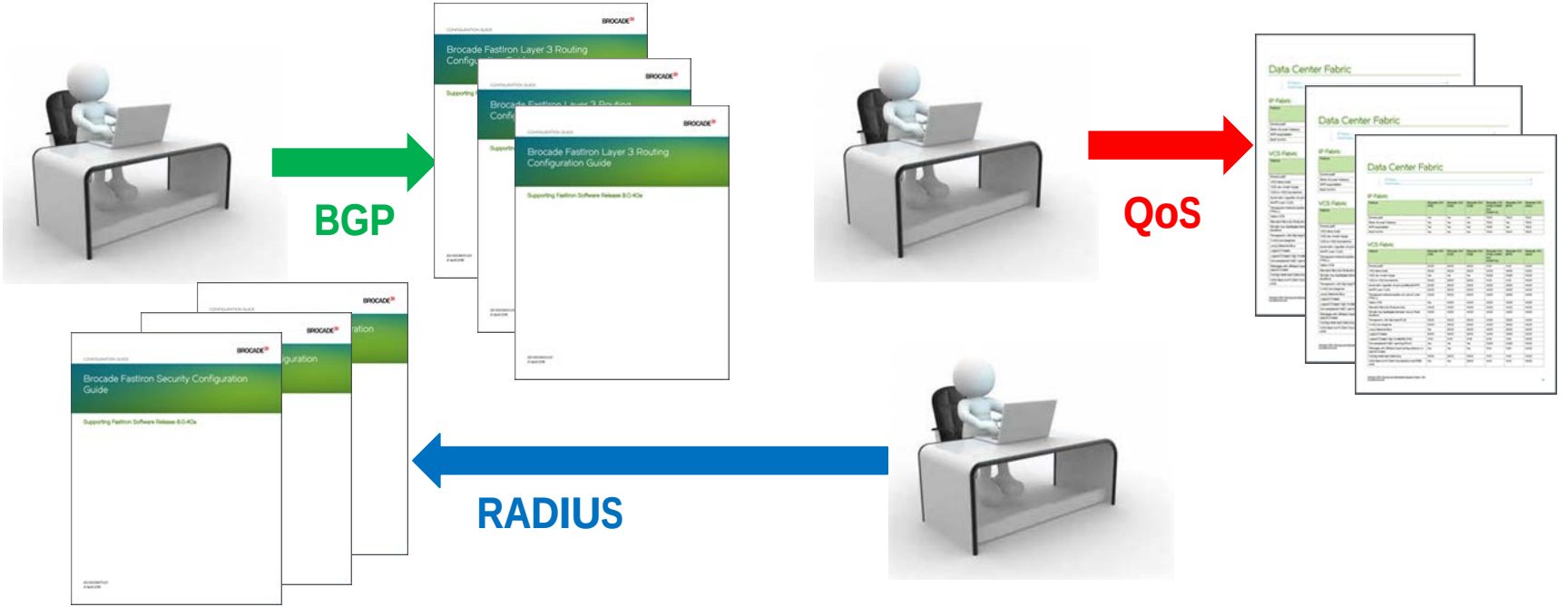
Writes QoS, BGP, RADIUS

Platform C books

- **Each technology on three platforms has a different writer**
- **Overlapping content, inefficient, three edit runs, no reuse**

After:

Technology-centric content developers

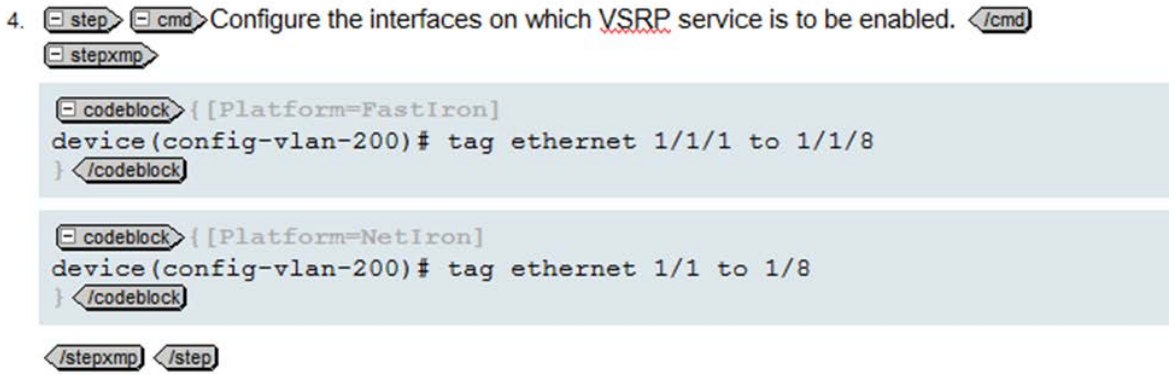


- **Our content developers "own" technologies across all platforms**
- **Benefits seen in writer expertise, efficiency, and reuse possibilities**

Reuse Mechanics

- Analyze the content across platforms to find the reuse; write generically
- Work with Subject Matter Experts (SMEs) to reorder content – did we just add features to the end of a chapter?
- Control your reuse by using a library for boilerplate content, but keep other content in context. We use platform release and technology folders.
- Use conditions to cope with different command syntax, prompts, or exceptions.

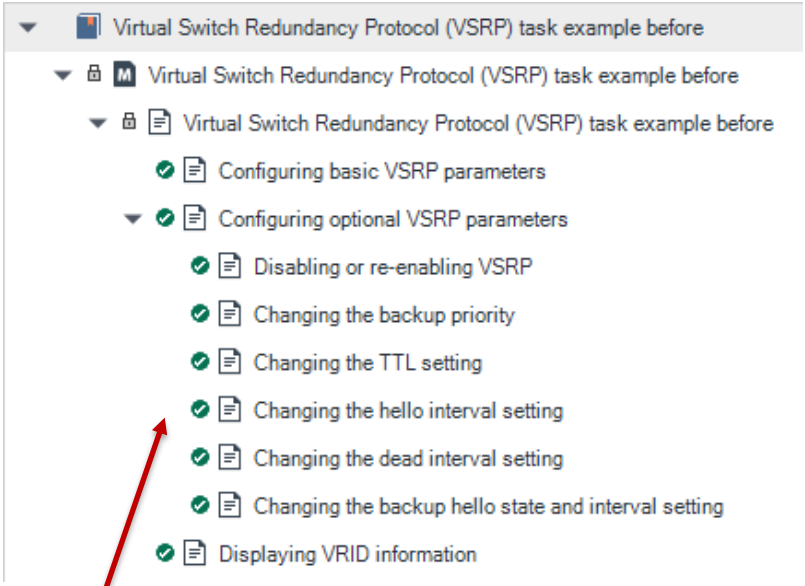
For Example:



Addressing Quality - Reworking Content

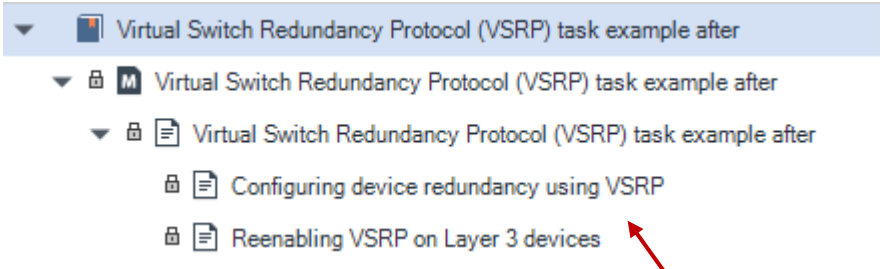
Task-based configuration content

Existing content with one-command topics



Each topic contains one command with several pages of output for **show** commands. Customers have to read many pages to gather the syntax for one task.

New task-based (use cases) content



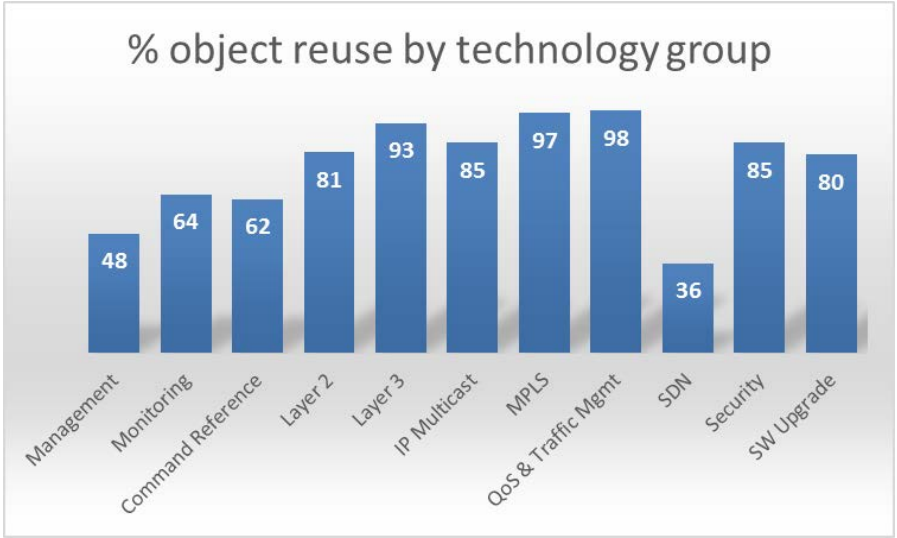
Each topic is a customer use case with numbered steps. Each step has the appropriate command syntax or output.

Separate command pages are now in the command reference with full syntax and examples or **show** command output and field descriptions.

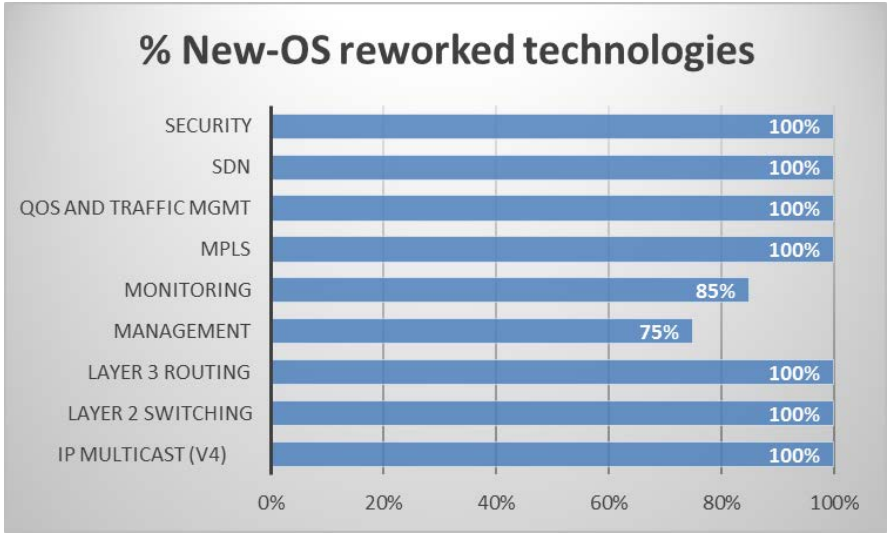
Result is leaner configuration guides.

Gather metrics to validate your content reuse

Reuse:



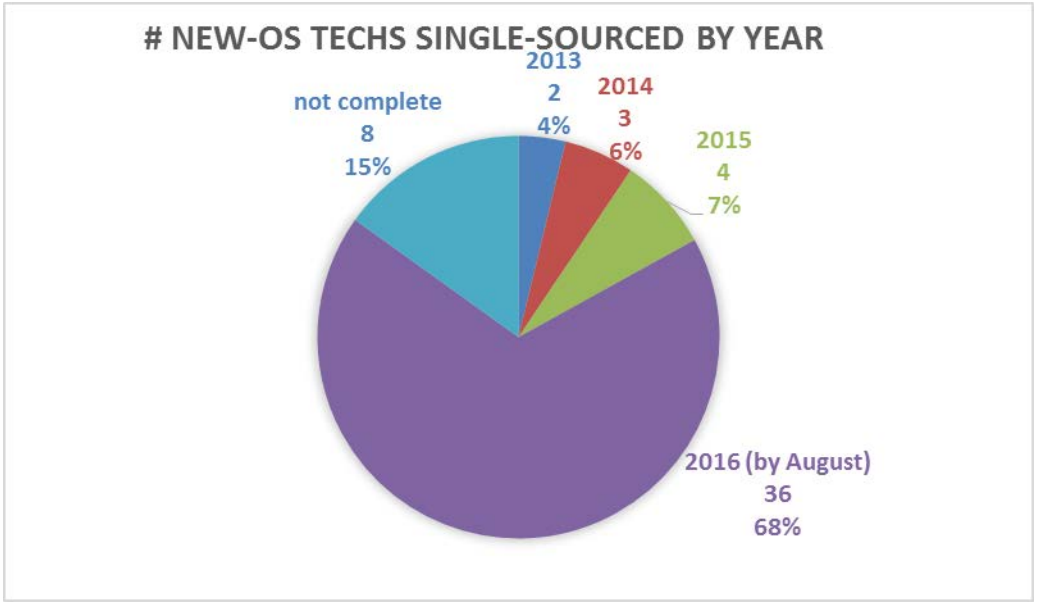
Quality:



Report using unit of reuse. Avg. content reuse across these tech groups = 75 %, with avg. # objects per book = 313

We committed to reworking 26 priority techs, achieved 51 techs

Progress of single-sourcing by year



Total # reusable techs=53

Single-sourcing content across at least 2 OS platforms/products

Note: Not all technologies are reusable or supported across all platforms

Challenges We Faced

From Writers:

I have to write features for four platforms? Attend 4 platform meetings?

From Engineers:

Where is my content from the old Admin Guide?

From Product Marketing:

We want a new book for our new technology, we don't want it in the xyz book!

From Management:

We're much more efficient, but are we improving quality? Do we have metrics that can highlight our quality improvements?

Expectations: How did we do?

Expectation 1: (No more writers)

- By creating technology owners, we became more efficient and knowledgeable about our technologies. We have delivered one extra doc set and are working on a second.

Expectation 2: (All content is written once and reused as is)

- We have been able to reuse most content across at least two platforms, and some content across four platforms. Most content was updated in this process and future updates should be minimal.

Expectation 3: (Don’t want to see other platforms mentioned)

- We used generic wording and use cases. Added conditions for platform exceptions.

Expectation 4: (Higher quality docs)

- We reworked content to add use cases (tasks with steps) as we single-sourced.

The Journey Continues…

Our journey was not as dramatic as Pip's life, but we faced great expectations from Engineers, Management, and Product Marketing.

We met and managed expectations by determining our structure, deciding our unit of reuse, reassigning writers as technology owners, and reworking content.

BROCADE

Questions?

Lelia Wright

September 13, 2016