

Using Distributed Version Control Systems: Enabling enterprise scale, XML based information development



Introduction

- Dr. Adrian R. Warman
Information Architect
IBM United Kingdom Limited
Hursley Park
Winchester
Hampshire
SO21 2JN

mail: Adrian.Warman@uk.ibm.com

Twitter: @warmana

Google+: Adrian Warman

Tel: +44-196-281-9176

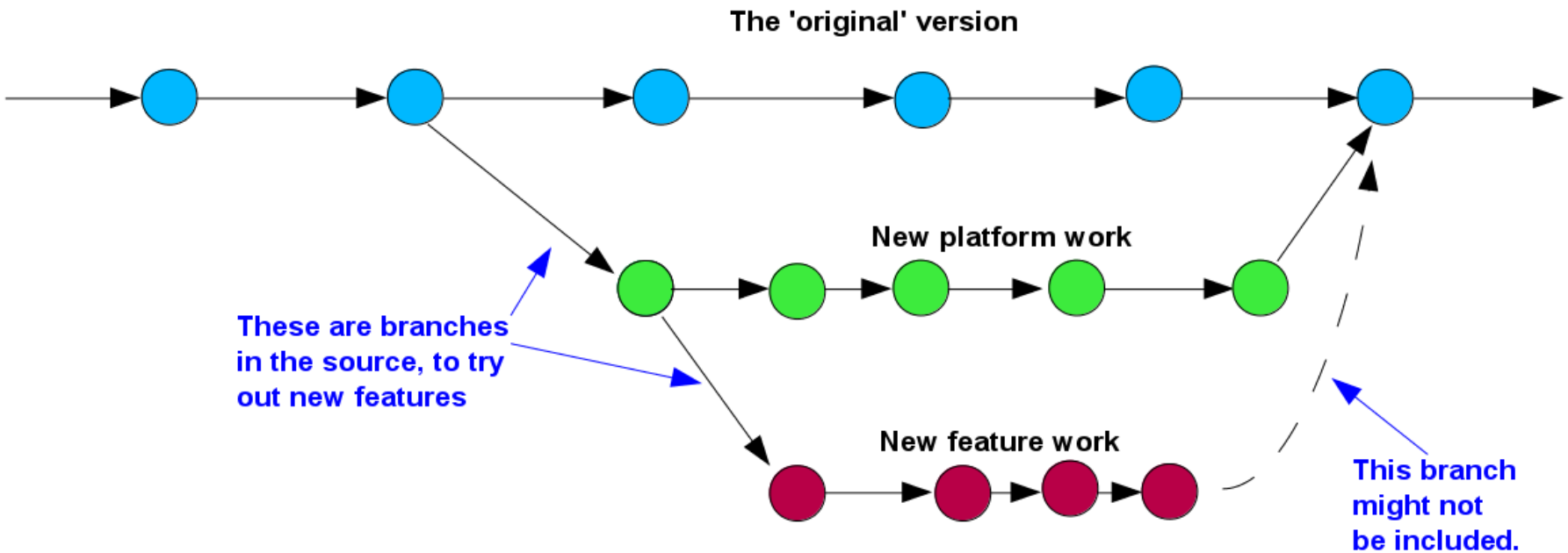
Disclaimer

- Any views or opinions expressed in this presentation are those of the author, and do not necessarily represent official positions, strategies or opinions of International Business Machines (IBM) Corporation.
- No guarantees are offered as to the timeliness, accuracy or validity of information presented.

Definitions

- Information Development
 - Enterprise documentation characteristics
- Information Set
 - Information stream
- Darwinian Information Typing Architecture
 - XML Markup
- Distributed Version Control System

Workflow



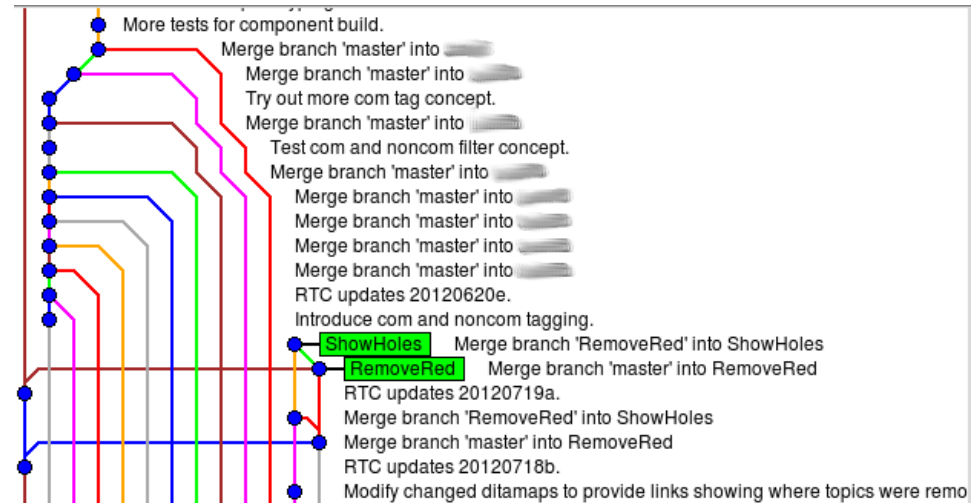
Basic XML content creation using DITA

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC
  "-//OASIS//DTD DITA Map//EN" "map.dtd">
<map xml:lang="en-us">
  <title>Getting started</title>
  <topicref
    href="quickstartguide/exploring-the-
dita-ot.dita"
    collection-type="sequence">
    <topicref
      href="readme/installing-full-
easy.dita"/>
    <topicref
      href="quickstartguide/rundemo.dita"/>
    <topicref
      href="quickstartguide/runmore.dita"/>
  </topicref>
</map>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE task PUBLIC
  "-//OASIS//DTD DITA Task//EN" "task.dtd">
<task id="exploring-the-dita-ot">
  <title>Getting Started with the DITA Open
Toolkit</title>
  <shortdesc>The <ph><cite>Getting Started
Guide</cite></ph> is designed to provide a
guided exploration of the DITA Open Toolkit.
It is geared for an audience that has little
or no knowledge of build scripts or DITA-OT
parameters. It walks the novice user through
installing the full-easy-install version of
the toolkit and running a prompted
build.</shortdesc>
</task>
```

DVCS Principles

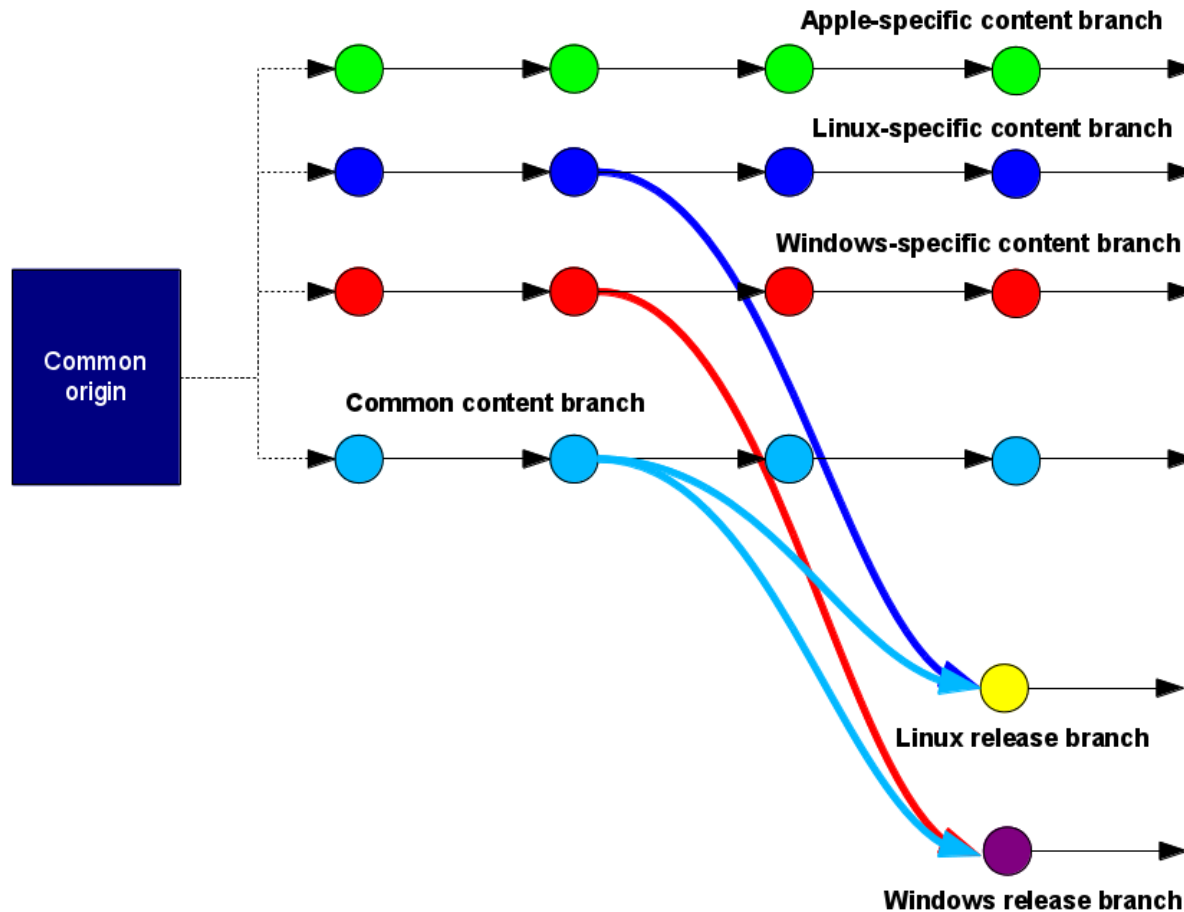
- Contrast with VCS
- More than one person working on a documentation set
- Storing versions of files in a repository
- Everyone has a complete copy of the entire repository
- Local changes can be shared
- Merging should be as automatic as possible
 - But fail gracefully if necessary



Storage requirements for a local repository

- Subversion (VCS)
 - Server copy of entire repository = 800+ MB
 - Local copy of repository and snapshot = 1.6 GB
- Git (DVCS)
 - Server copy of entire repository = 331 MB
 - Local copy of repository and snapshot = 1 GB

Using a DVCS for multiple branches and multiple deliveries



Advantages and disadvantages

- Not a Content Management System (CMS)
- Mechanisms for ensuring validity of markup
- Re-use is possible
 - Simplified by branch merging?
- Assembly of content on demand
- Multi-branch management can become complex
 - Very complex
- Do local storage requirements scale?
- Going beyond intent of original DVCS concept
- 'Moving' sub-content might be problematic

Summary, Q&A

- Basic characteristics of Enterprise documentation
- Similarity to Enterprise software development
- Application of DVCS solution to some challenging documentation tasks
- The investigation continues...