DevOps meets Docs: Documentation as Code

OSSummit Europe 2018

Robert Krátký, У @rkratky Principal Technical Writer, Red Hat



IN THIS PRESENTATION

1. What does 'Docs as Code' mean?

- a. What is it?
- b. What's involved?

2. Benefits

- a. For writers and developers
- b. For organizations or projects

3. Challenges

- a. For writers and developers
- b. Technical



WHAT DOES 'DOCS AS CODE' MEAN?

Writing, testing, publishing, and maintaining documentation using the same tools developers use for software code.

Goals

- Content authoring
- Formatting, styling
- Version control
- Issue tracking
- Testing, validation
- Publishing

Tools

- → Text editors
- → Markup languages
- \rightarrow Git, SVN, Mercurial, ...
- → JIRA, BugZilla, Mantis, ...
- → Scripts, linters, spell-checks
- → Site builders, CMS, ...

CONTENT AUTHORING

→ TEXT EDITORS

- Writers have freedom of choice
- Advanced batch operations
- Using external tools to process text



FORMATTING, STYLING → MARKUP LANGUAGES

- Universal readability (plain text)
- Syntax validation
- (More or less) semantic, i.e. parseable

VERSION CONTROL

→ GIT AND CO.

- Distributed, decentralized (Git)
- Integral peer-review workflows
- Docs can share space with code
- Public hosting & on-premise deployment

https://github.com/redhat-documentation/

ISSUE TRACKING

→ BUGZILLA, JIRA, ...

- Integration with version control
- API hooks (e.g. generating 'release notes')
- Agile facilitation

TESTING, VALIDATION → SCRIPTS, LINTERS

- Continuous integration
- Customizable, extendable
- Link validity, component checking
- Semantic syntax parsing

• Continuous deployment

- Local and on-production building
- Containerized toolchains (multi-platform)

BENEFITS

FOR WRITERS & DEVELOPERS

- Developers familiar with toolchain
 - Contributions from developers
 - Fly-by contributions
 - Community contributions
- Integration with developer teams
 - Involved with product planning
 - Included in the **definition of done**
 - Easier collaboration with QA
 - Sense of common responsibility

BENEFITS

FOR ORGANIZATION, PROJECT

- No vendor lock-in
 - Future-proof format & tools
 - Easy switching to different formats & tools
- Tighter collaboration between teams, depts.
 - Unified, streamlined workflows
 - Efficiency (automation)
 - Aligning releases with publishing

CHALLENGES

FOR WRITERS & DEVELOPERS

• Resistance from writers

- Training for writers may be needed
- Steep learning curve
- Many tools with single-purpose function (vs monolithic docs system) - tools may seem too difficult, yet simple
- Resistance from developers
 - Documentation slows down 'done'
 - Corrections more visible; hurt pride

CHALLENGES

TECHNICAL

- Content preparation
 - (Potentially) non-trivial conversion
 - Release disruption
- Technical support & administration (depending on scope)
 - IT resources
 - IT staff

DISCUSSION

- Your experience?
- Tools of choice?
- Development models?
- When adopting a new system
 - Convert existing?
 - Start from scratch?
- Metadata?
- Peer-review approaches?

THANK YOU

rkratky@redhat.com

