BRIDGING THE GAP BETWEEN LEGACY DOCS AND MODULAR CONTENT

FOSDEM 2017

Robert Krátký, Senior Technical Writer
rkratky@redhat.com
IN THIS PRESENTATION

1. Intro, terminology
   a. Why bother?
   b. What’s “legacy”?
   c. What’s “modular”?

2. What’s the problem?
   a. Too much docs
   b. Hard to find
   c. Hard to navigate

3. What’s our solution?
   a. Document only what users really need
   b. Facilitate sharing & single-sourcing
   c. Modular presentation

4. Discussion
INTRODUCTION

- Why? -- Landscape is changing:
  - Software **packaging**
    - Monolithic distros → containerized apps
  - Software **delivery**
    - Release cycles → continuous deployment
TERMINOLOGY

- **Legacy docs** = big guides/books
  - comprehensive
  - feature-based

- **Modular docs** = lean, concise articles (units)
  - targeted
  - user-story-based
FEATURE-BASED VS USER-STORY-BASED

FEATURES
Describe all available features; comprehensive

1. Using Onions
   1.1. Peeling Onions
   1.2. Cutting Onions
   1.3. Frying Onions
2. Using Eggs
   2.1. Breaking Eggs
   2.2. Mixing Eggs with Milk
   2.3. Frying Eggs
## FEATURE-BASED VS USER- STORY-BASED

### FEATURES
Describe all available features; comprehensive

1. Using Onions
   1.1. Peeling Onions
   1.2. Cutting Onions
   1.3. Frying Onions
2. Using Eggs
   2.1. Breaking Eggs
   2.2. Mixing Eggs with Milk
   2.3. Frying Eggs

### USER STORIES*
Explains how to achieve specific goals; selective

1. Making an Onion Omelette
   1.1. Setting up Workplace
   1.2. Preparing Ingredients
   1.3. Selecting Spices
   1.4. Combining Ingredients
   1.5. Frying Omelette Mixture
   1.6. Serving the Omelette
   1.7. Other Onion Recipes

*As a $(type of user), I want to $(goal) so that $(reason).

Questions?
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:

- Difficult to maintains
  (nobody likes to write docs)
- Lots of duplication
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:

- Difficult to maintain (nobody likes to write docs)
- Lots of duplication
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:

- Difficult to maintains (nobody likes to write docs)
- Lots of duplication

Hard to find:

- Which guide explains what?
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:

- Difficult to maintains (nobody likes to write docs)
- Lots of duplication

Hard to find:

- Which guide explains what?

  Content rot

  Google it instead & go to SO
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:

- Difficult to maintain (nobody likes to write docs)
- Lots of duplication

Hard to find:

- Which guide explains what?

Hard to navigate:

- Guides are too long (scrolling, searching, saving...)
- Experience not customizable
WHAT’S THE PROBLEM (with legacy docs)?

Too much content:
- Difficult to maintain (nobody likes to write docs)
- Lots of duplication

Hard to find:
- Which guide explains what?

Hard to navigate:
- Guides are too long (scrolling, searching, saving...)
- Experience not customizable

Questions?
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate **user stories**
- Cut down on the amount of content
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate **user stories**
- Cut down on the amount of content

Efficient use of resources
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate user stories
- Cut down on the amount of content

Efficient use of resources

Facilitate sharing & single-sourcing:

- Avoid duplication
- Bring together upstream & downstream
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate **user stories**
- Cut down on the amount of content

Facilitate sharing & single-sourcing:

- Avoid duplication
- Bring together upstream & downstream

Efficient use of resources

Save time & bring devs on board
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate **user stories**
- Cut down on the amount of content

Facilitate sharing & single-sourcing:

- Avoid duplication
- Bring together upstream & downstream

Modular writing & presentation:

- **Modules form assemblies**
- “Curated” assemblies, “ad hoc” docs
WHAT’S OUR SOLUTION?

Document only what users really need:

- Identify and validate **user stories**
- Cut down on the amount of content

Facilitate sharing & single-sourcing:

- Avoid duplication
- Bring together upstream & downstream

Modular writing & presentation:

- *Modules form assemblies*
- “Curated” assemblies, “ad hoc” docs

Questions?
ASSEMBLIES & MODULES

Documentation realization of a user story

“topic” -- too ambiguous; not used on purpose
ASSEMBLIES & MODULES

- **Title**: action-oriented, specific, e.g. “Making an Onion Omelette”.

- **Purpose**: What is going to be accomplished.

- **Prerequisites**: Actions that need to be taken or conditions that need to be satisfied.
ASSEMBLIES & MODULES

Title
What is explained, e.g. “Understanding the Importance of Omelettes in French Cuisine”.

Explanation
Concise, to the point.

Additional resources
Where to go for further details or background info.
ASSEMBLIES & MODULES

ASSEMBLY

INTRODUCTION

CONCEPT MODULE

PROCEDURE MODULE #1

PROCEDURE MODULE #2

Title
“Frying Omelette Mixture”.

(Purpose)
(Prerequisites)

Steps
Commands, i.e.
1. Select ...
2. Open ...
3. Copy ...
4. ...

Additional resources
ASSEMBLIES & MODULES

ASSEMBLY
- INTRODUCTION
- CONCEPT MODULE
- PROCEDURE MODULE #1
- PROCEDURE MODULE #2
- ADDITIONAL RESOURCES & REFERENCES

Links & references to:
- Related assemblies
- Individual modules
  - Concepts
  - References
  - Related procedures
- Websites
- Manual pages
- Other onion recipes
- ...

Questions?
WRAP UP

LEGACY
Long, feature-based guides, going after completeness

MODULAR
Adapting into modular, action-oriented units, based on user stories
WRAP UP

LEGACY
Long, *feature-based* guides, going after *completeness*

Lots of content, hard to maintain, *learning curve* for contributors *too steep*

MODULAR
Adapting into modular, *action-oriented* units, based on *user stories*

Cut down on the *amount of content*, small chunks = low-hanging fruit for new people
WRAP UP

LEGACY
Long, feature-based guides, going after completeness
Lots of content, hard to maintain, learning curve for contributors too steep
Cumbersome navigation, static content, no customization

MODULAR
Adapting into modular, action-oriented units, based on user stories
Cut down on the amount of content, small chunks = low-hanging fruit for new people
Hierarchical, metadata-based navigation, dynamic presentation

Questions?
DISCUSSION

- Do you use something similar?
- What are the major risks and/or caveats of this approach?
  - No clear content strategy
    - tons of assemblies
    - no real order
      ➔ mind mapping?
  - Initial overhead
    - converting existing docs
    - maintenance of legacy docs
      ➔ work progressively; while fixing old docs, adapt into modular
- What are your major grievances with docs (and open-source docs)?
- ...
THANK YOU

rkratky@redhat.com