

# **GOING MODULAR: TURNING LEGACY DOCS INTO USER STORY-BASED CONTENT**

TW Exchange of Experience, 2017, Prague

Aneta Šteflová, Robert Krátký  
aneta@redhat.com, rkratky@redhat.com

# IN THIS PRESENTATION

## **1. Terminology**

- a. What's "legacy" & "user story-based"?

## **2. What's the problem?**

- a. Too much docs
- b. Hard to find and navigate

## **3. What's our solution?**

- a. Document only what users really need
- b. Modular writing

# TERMINOLOGY

- **Legacy docs @ Red Hat** = big guides/books
  - comprehensive
  - **feature-based**
- **User story\*-based docs** = lean, concise articles (units)
  - targeted
  - (ideally) modular

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

---

# FEATURE- VS USER STORY-BASED

## FEATURES

Describe all available features; comprehensive

1. Using Onions
  1. Peeling Onions
  2. Cutting Onions
  3. Frying Onions
2. Using Eggs
  1. Breaking Eggs
  2. Mixing Eggs with Milk
  3. Frying Eggs

## USER STORIES\*

Explain how to achieve specific goals; selective

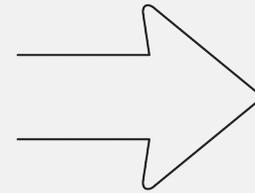
1. Making an Onion Omelette
  1. Setting up Workplace
  2. Preparing Ingredients
  3. Selecting Spices
  4. Combining Ingredients
  5. Frying Omelette Mixture
  6. Serving the Omelette
  7. Other Onion Recipes

*\* As a novice cook, I want to make an onion omelette so that I can surprise my family for breakfast.*

# THE PROBLEM (with legacy docs)

## Too much content:

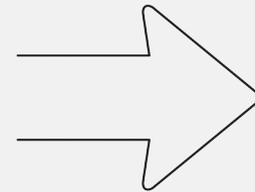
- Difficult to maintain
- Lots of duplication



Content  
rot

## Hard to find and navigate:

- Which guide explains what?
- Need to piece info from many places



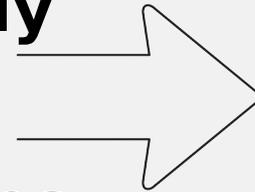
Google it  
instead &  
go to SO



# WHAT'S OUR SOLUTION?

## **Document only what users really need:**

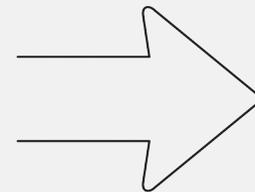
- Identify and validate **user stories**
- Cut down on the amount of content



Efficient  
use of  
resources

## **Modular writing:**

- *Assemblies* made of *modules*
- Templates make writing easier
- Modular pieces easier to chew off

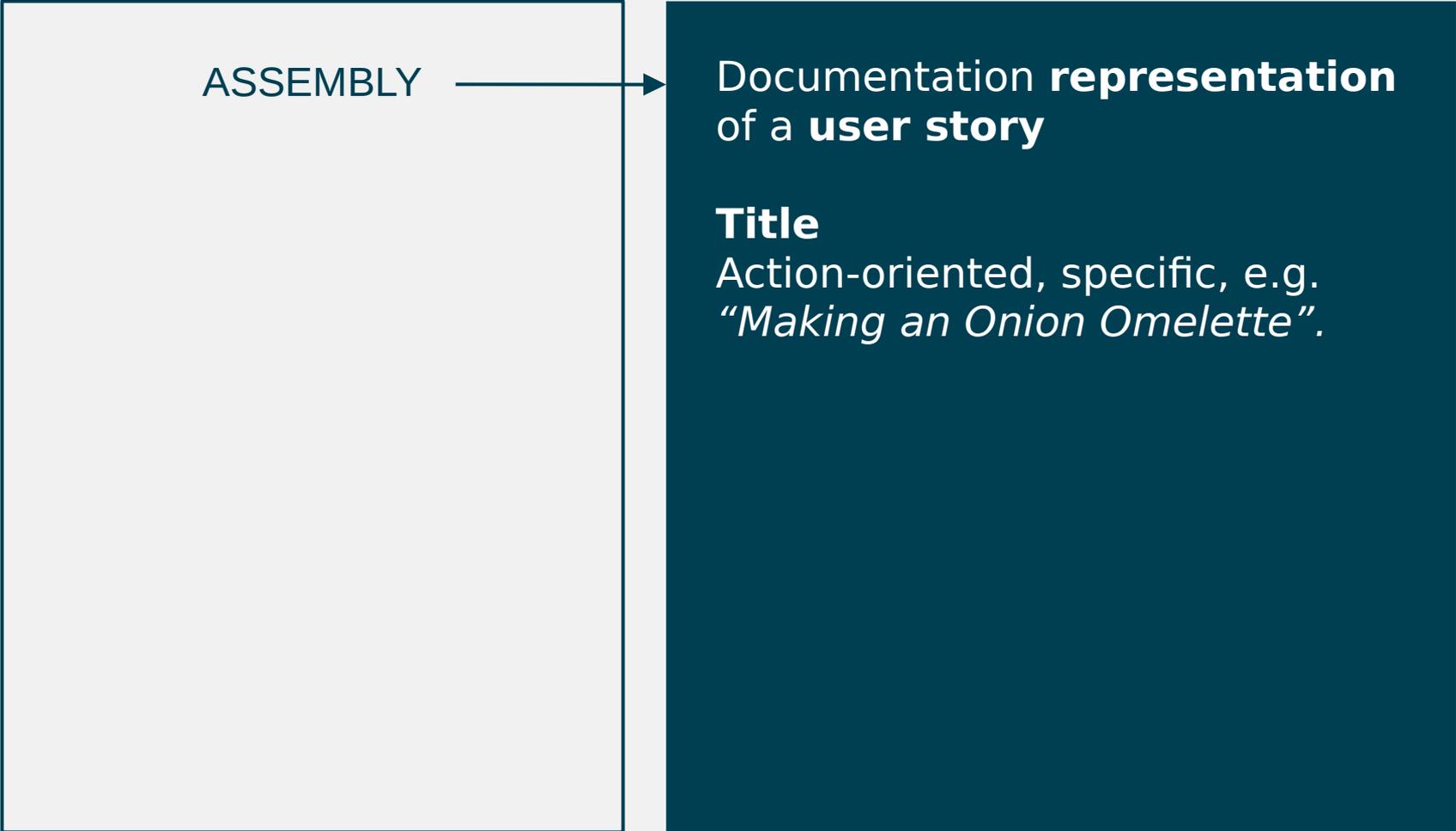


Save time  
&  
encourage  
new  
people



# ASSEMBLIES & MODULES

ASSEMBLY

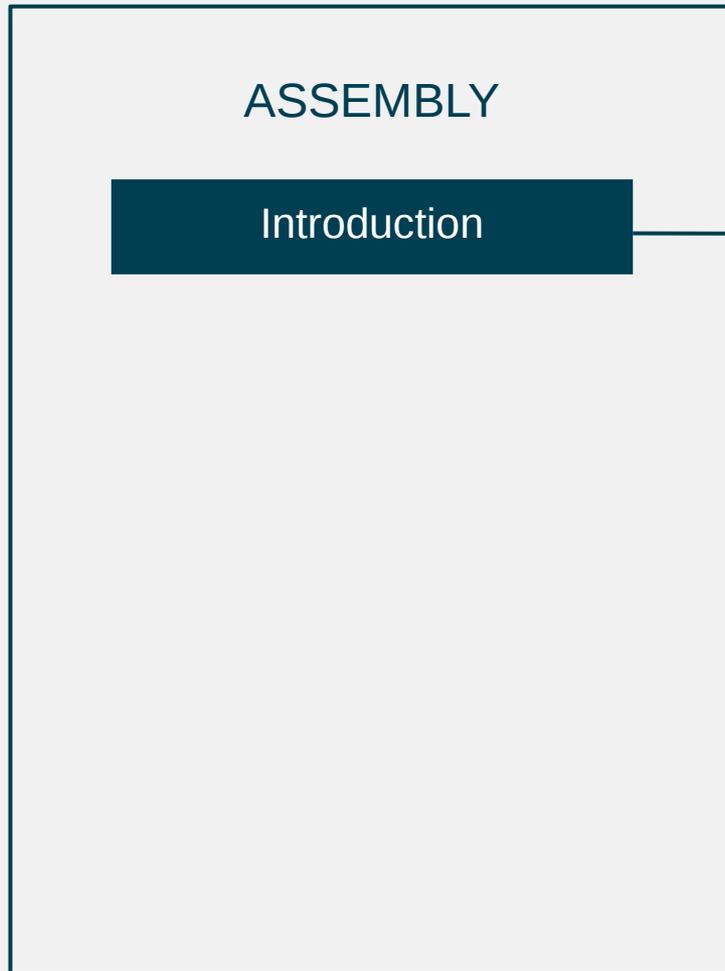


Documentation **representation**  
of a **user story**

## **Title**

Action-oriented, specific, e.g.  
*“Making an Onion Omelette”*.

# ASSEMBLIES & MODULES



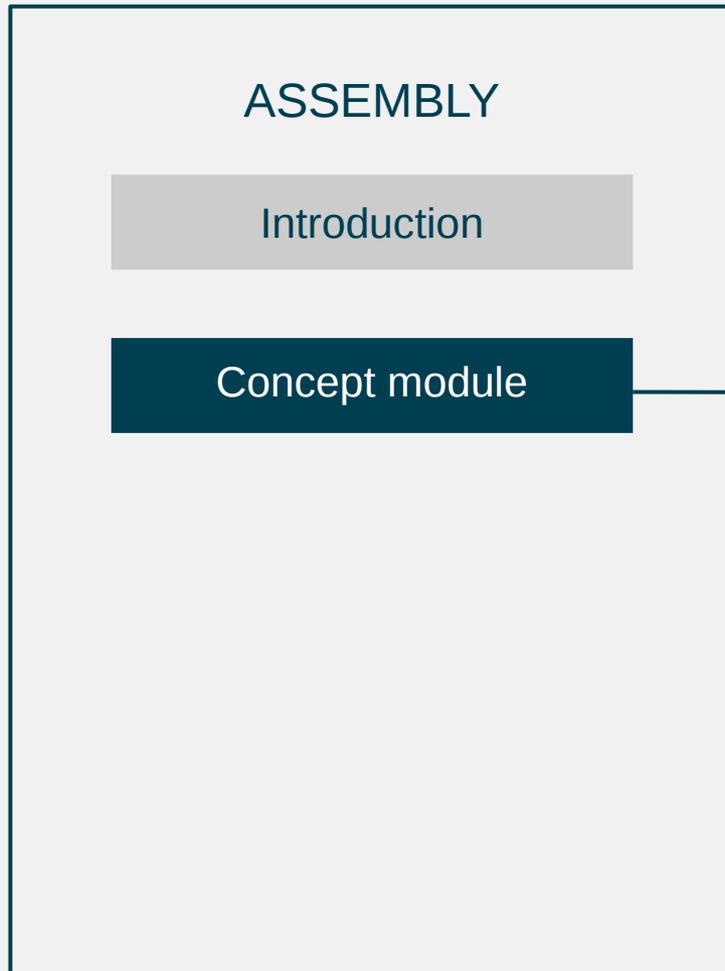
## **Purpose**

What is going to be accomplished.

## **Prerequisites**

Actions the user needs to take.  
Conditions the user needs to satisfy.

# ASSEMBLIES & MODULES



## **Title**

What needs to be explained, e.g. *“Understanding the Importance of Omelettes in French Cuisine”*.

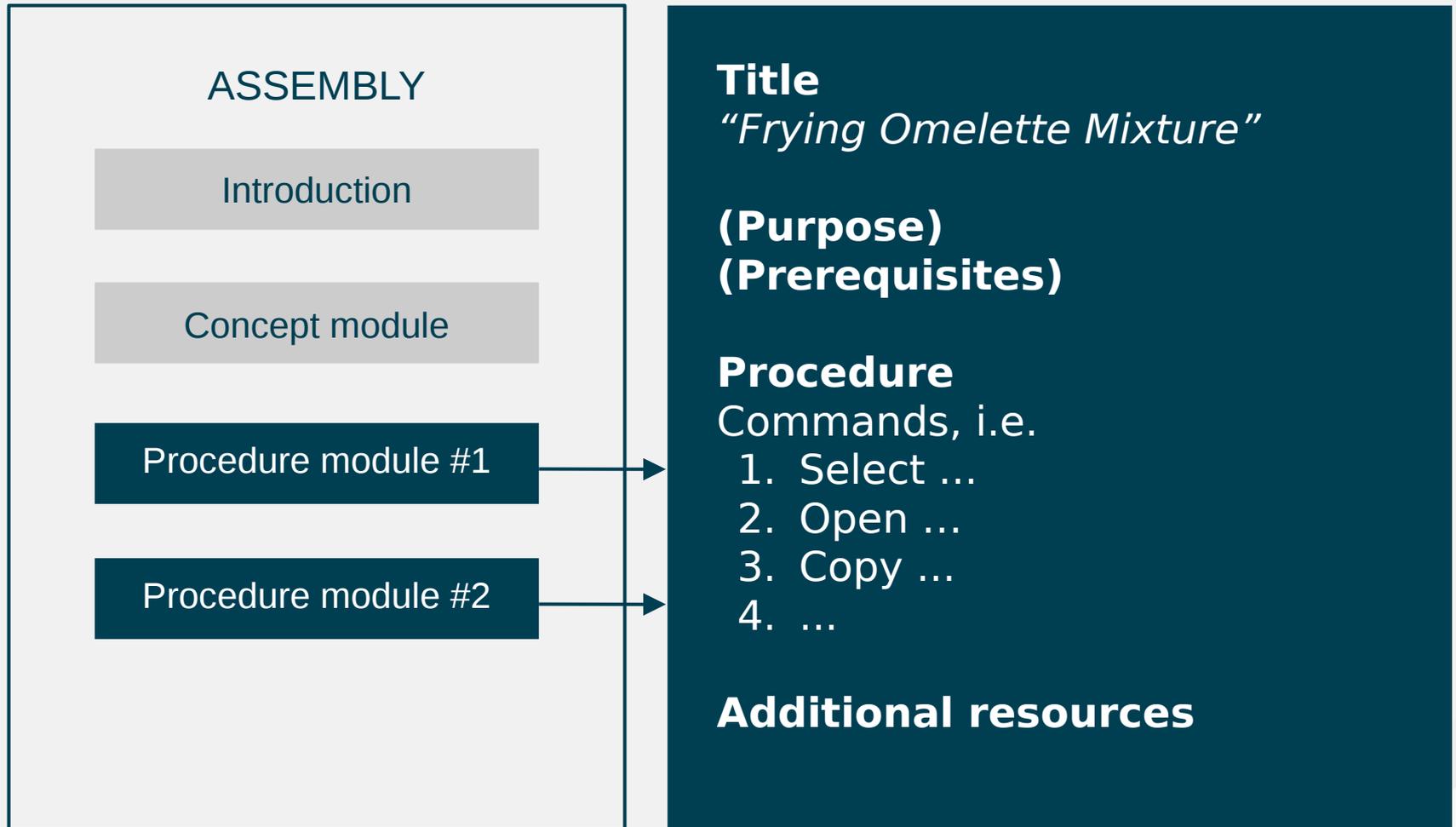
## **Concept explanation**

Concise, to the point.

## **Additional resources**

Where to go for further details or background info.

# ASSEMBLIES & MODULES



# ASSEMBLIES & MODULES

## ASSEMBLY

Introduction

Concept module

Procedure module #1

Procedure module #2

Additional resources

## Links & references to:

- Related assemblies
- Individual modules
  - Concepts
  - References
  - Related procedures
- Websites
- Manual pages
- Other onion recipes
- ...

# 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



## USER STORY-BASED

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

**Easier navigation**, potential for reuse and metadata-based filtering

# HOW WE MAKE THIS HAPPEN #1

- Getting user stories
- New docs based on user stories
- Auditing legacy docs

# HOW WE MAKE THIS HAPPEN #2

Knowledge-sharing among writers:

- Internal presentations
- Templates
- Pair writing
- Workshops



# WANT TO KNOW MORE?

Robert Krátký wrote an article about user story-based docs:

<https://opensource.com/article/17/6/documentation-based-user-stories>