This Is a Test Slide

- Can you read this text?

Sometimes It's Just Hard

Why Some DITA-OT Features Remain Out of Reach

1. DITA-OT in a Jar
2. Incremental Publishing

The Problem

- Most things are possible if you have a lot of resources and a lot of time
  - Most open source projects have neither
- Most things are technically possible
  - Some things are more complex than others

DITA-OT in a Jar

- DITA-OT requires a directory on disk
- Java code is usually distributed as JAR files that don't require a directory structure
- A single JAR file would be easier to embed and even compile into an executable

DITA-OT in a Jar

1. Ant assumes you're working with files on disk
2. Existing plug-ins assume you can read files from plugin directory
3. Plug-in extension points are evaluated at integration time and that creates files

DITA-OT in a Jar

- Write plug-in integration files to an application data directory or do integration at run-time
- Support plug-ins that are themselves JAR files in a new plug-in format
- Supporting two plug-in formats is extra work and may never see adoption

Incremental Publishing

- Compared to modern SSGs DITA-OT is slow
- DITA-OT is used by large enterprises
- CI/CD systems are common

Incremental Publishing

- DITA processing is more complex than some other markup languages
- Existing plug-ins can extend preprocessing
- Incremental processing is effectively cache invalidation

Incremental Publishing

- To invalidate proprocess cache, we need to track which files can affect each other
- For example, if
    - key text changes
    - topic title changes
    - topic inside chunk changes
- Invalidate topics cascade their invalid state

Incremental Publishing

- Requires a place to store the cache in CI systems
- Requires opt-in from plug-ins

Done