*instinctools™

Experts in Information Management Solutions and Services

Innovative Information Management

Documentation of Eclipse Applications with DITA

Eclipse Embedded Day Stuttgart 2010

Gunthilde Sohn, instinctools GmbH

Agenda

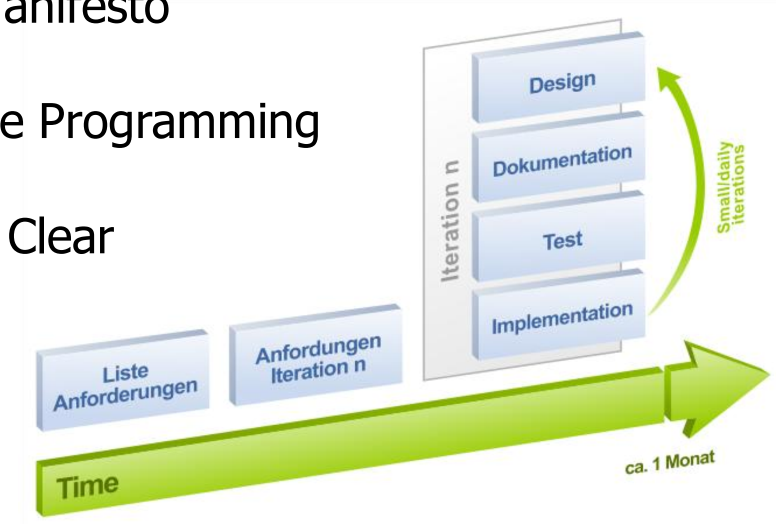
- Challenges in Software documentation
- Single-source documentation with DITA
- DITAworks as a tool option
- Software development and documentation hand in hand
- Demo
- Q&A
- More information

Challenges in Software Documentation

Agile Process Model

Changing market requirements

- Agile Manifesto
- Extreme Programming
- Crystal Clear



✓ „Functioning programs more important than detailed documentation „

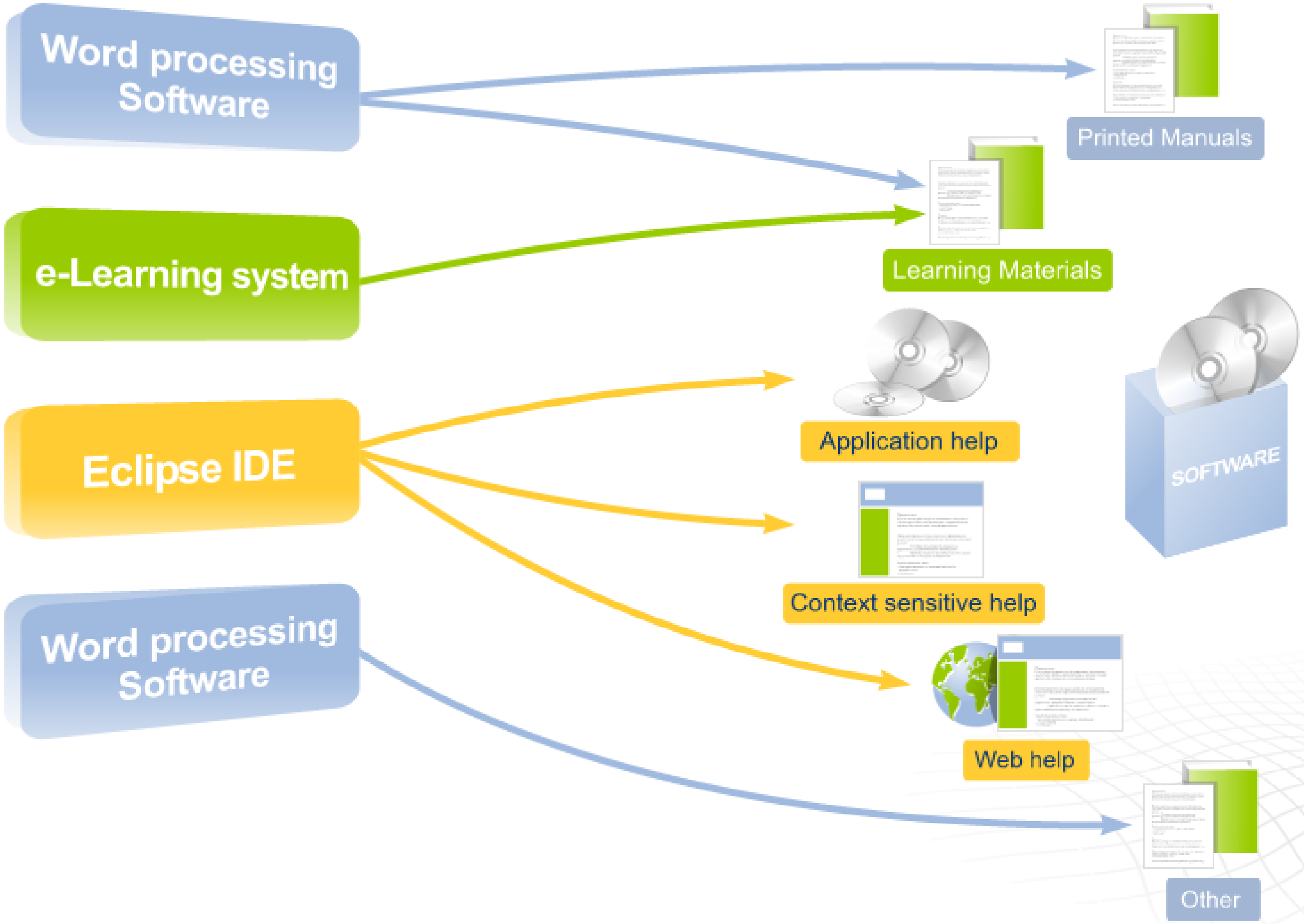
Project specifics

- changing requirements
- changing, incomplete targets
- small teams
- short release cycles
- more informal organisation
- flexibility, speed

✓ Documentation to ensure next steps

✓ Development and documentation more synchronized

Traditional Documentation Toolset



Typical SW documentation deliverables:

- ✓ Printed documentation materials (manuals, references etc.)
- ✓ Application help
- ✓ Context sensitive help
- ✓ Documentation materials that should be published online (Online help)
- ✓ Training materials
- ✓ Reference documentation (API docs and s.o.)
- ✓ ... other documents

Most of these documents have quite a high potential for partial content reuse.

Challenges we are Facing in this Scenario are:

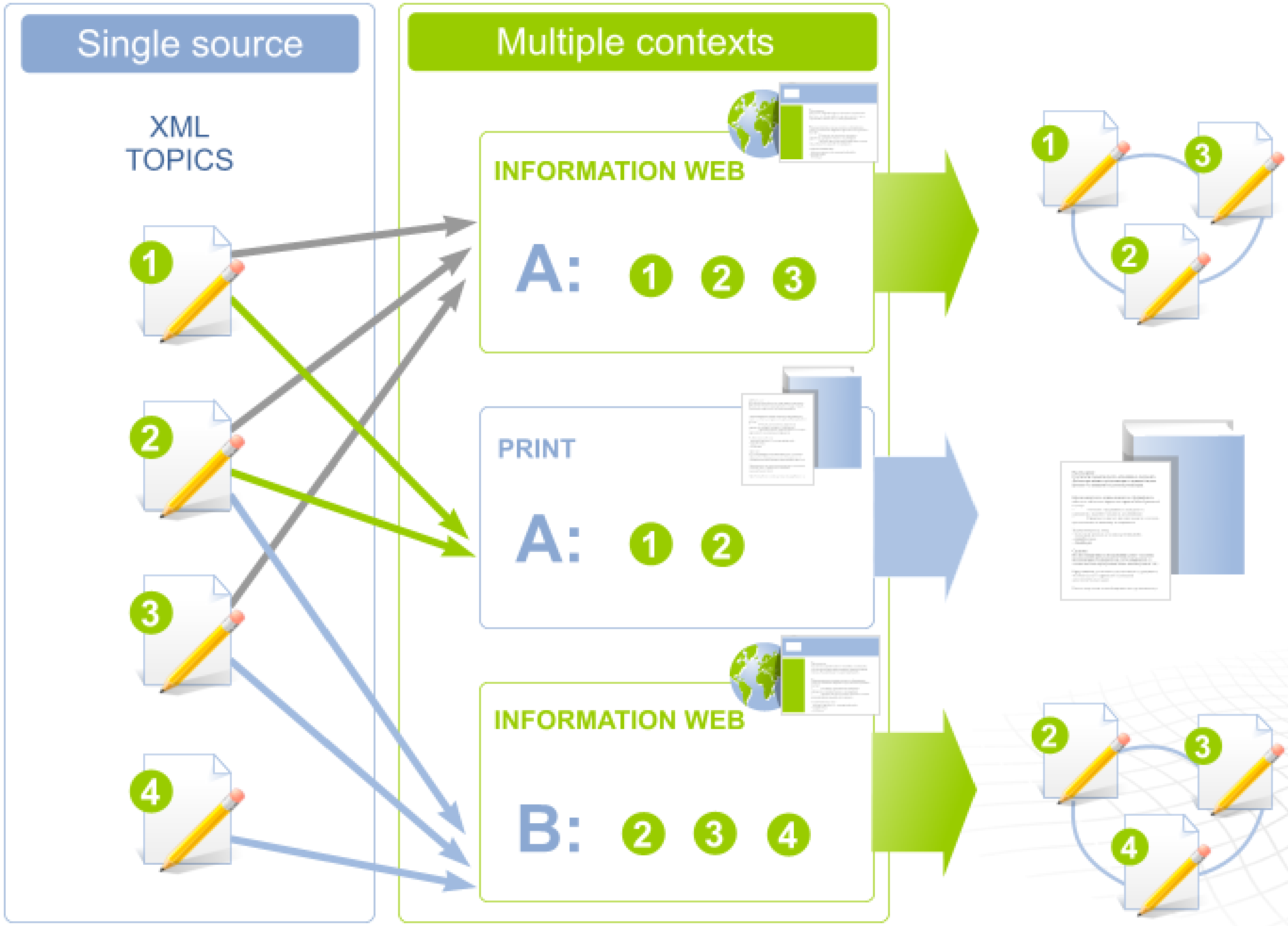
- ✓ Dealing with **different source formats** and **redundant content**
- ✓ Increasing **Complexity** of Documentation
- ✓ Globalization & Localization
- ✓ **High Quality** Expectations
- ✓ Different **Target Media**
- ✓ Need of **Integration**
- ✓ **Shortening** of Development Cycles
- ✓ Increasing Demand for **Documentation Variants**

<u>Conclusion:</u> Without **consistent documentation methodology** and appropriate **tool support** there is very less chance to manage all required deliverables efficiently

Solution: DITA

- ✓ **Single source publishing** as answer to these challenges.
- ✓ *Single sourcing allows:*
  - ✓ ***same content*** *to be used in* ***different documents*** *or in* ***various formats****.*
  - ✓ *labor-intensive and expensive work of editing only to be carried out* ***once, on one source document****.*
  - ✓ further **transformations to be performed mechanically**, by automated tools.
- ✓ One of the leading standards in XML single sourcing is **DITA:**
  - ✓ DITA stands for **Darwin Information Typing Architecture**
  - ✓ DITA was developed by IBM and donated to OASIS
  - ✓ DITA divides content into small, self-contained topics
  - ✓ DITA Topics can be reused in different deliverables.
  - ✓ Deliverables are modeled using special structure called **map.**

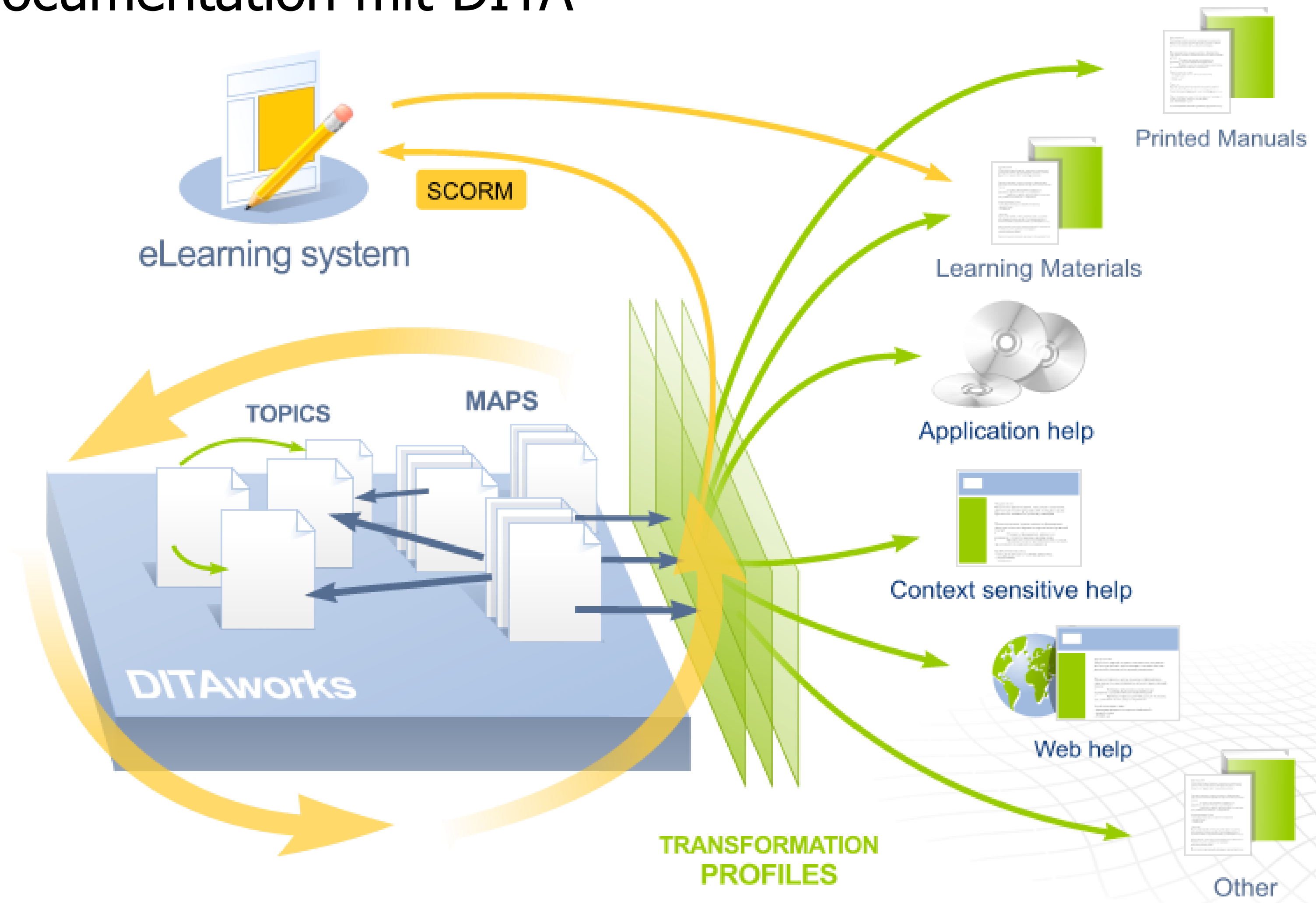
DITA Publication



DITA: Main features

- ✓ Topic-based DITA offers **flexibility in content organization**
- ✓ DITA is fairly **simple**. It uses about half as many tags as DocBook does
- ✓ DITA's modularity allows **reuse** (on **level of topic or single elements**)
- ✓ DITA allows **separation of content from presentation**
- ✓ DITA allows publishing into **different kinds of output formats**
- ✓ DITA enables use of **conditional content**
- ✓ DITA allows **process automation**
- ✓ DITA can be used **out-of-the-box**
- ✓ DITA allows you to **build on** solid base and saves time on:
  - ✓ New Infotype DTD/XML Schema development.
  - ✓ Style sheet development
- ✓ DITA is **extensible**. It enables new infotype development with backward compatibility to base DITA.

Documentation mit DITA



DITAworks as a tool option

DITAworks Product Core Areas

Modeling

Provides meta information for all DITAworks tooling and enables efficient use of specialization.

- **Import** of existing specializations from **DTD** and **XML Schema**
- **Export** of managed specializations to **DTD** and **XML Schema**
- **Visual editors** for DITA base model: Specialization editing
- Model **validations**
- **Extended model**:
  - Settings for editorial tools
  - WYSIWYG Style sheets
  - Custom validation rules
  - Custom transformations
  - etc.
- **Model refactoring**

Documentation Process

Provides a set of integrated editorial model-driven tools for efficient content authoring.

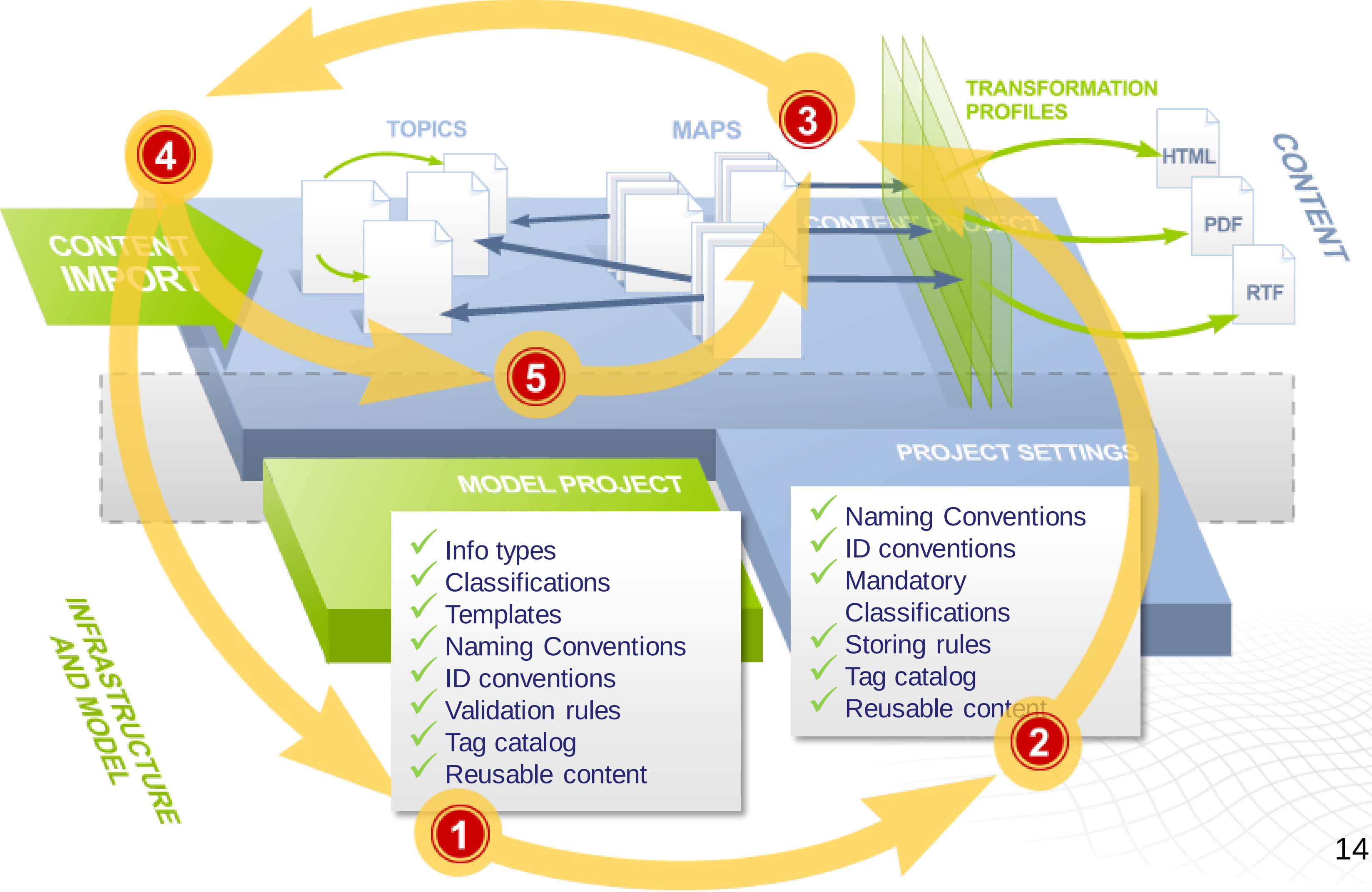
- **Content Import**
- **Infrastructural project rules**
- **Visual editors** for DITA topics and maps
- **Link management**
- **Content validation**
- **Content refactoring** (e.g. rename, move, content extraction etc.)
- **Reuse block management**
- **Tag management**
- **Versioning**
- **Collaboration**

Publishing & Export

Provides means for delivery to end-user .

- **Integrated OT** with some important improvements.
- **Publishing configurations** including settings for variables, conditional processing etc.
- Extended support of **Eclipse help**
- **Export** of content (full or partial)
- **Publishing server** for running headless transformations.

Getting Started with DITAworks: Agile Approach



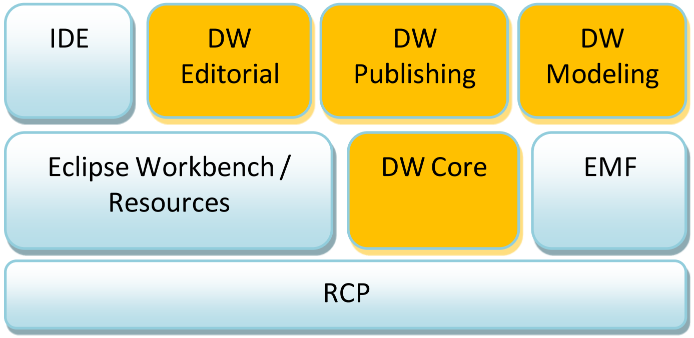
- ✓ Info types
- ✓ Classifications
- ✓ Templates
- ✓ Naming Conventions
- ✓ ID conventions
- ✓ Validation rules
- ✓ Tag catalog
- ✓ Reusable content

- ✓ Naming Conventions
- ✓ ID conventions
- ✓ Mandatory Classifications
- ✓ Storing rules
- ✓ Tag catalog
- ✓ Reusable content

DITAworks on Eclipse platform

Advantages:

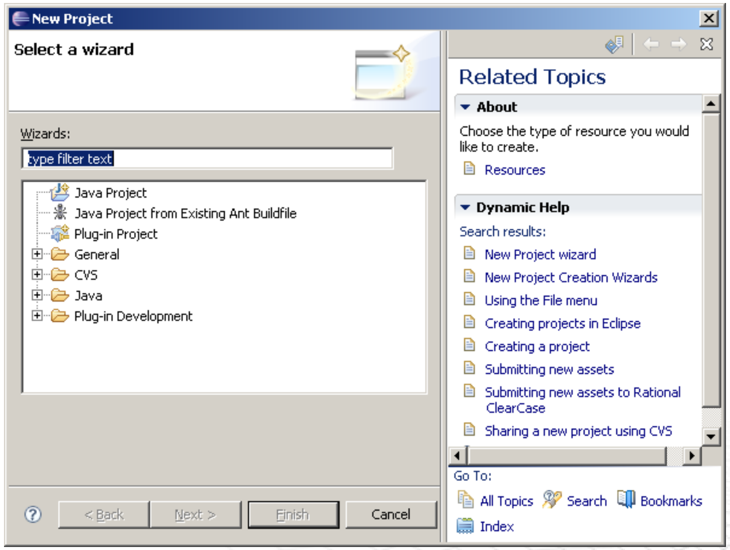
- ✓ Easy integration with other Workspace-oriented tooling (e.g. SW development tools)
- ✓ Easy integration with other Eclipse projects and technologies (Example: version control, task management with Mylyn and s.o.)
- ✓ Functional blocks can be used independently



What does Eclipse offer in the area of documentation?

Eclipse Help Subsystem:
Eclipse Plug-ins for helps integrated in Eclipse / RCP

- Eclipse Help can be used for:
  - Application help
  - Context sensitive help
  - Web/Online help (Infocenter)
- Eclipse Help Format is based on an XML toc that is referring to HTML files
- Eclipse IDE contains Plug-ins supporting authoring processes



New Project

Select a wizard

Wizards:

type filter text

- Java Project
- Java Project from Existing Ant Buildfile
- Plug-in Project
- General
- CVS
- Java
- Plug-in Development

Related Topics

About

Choose the type of resource you would like to create.

- Resources

Dynamic Help

Search results:

- New Project wizard
- New Project Creation Wizards
- Using the File menu
- Creating projects in Eclipse
- Creating a project
- Submitting new assets
- Submitting new assets to Rational ClearCase
- Sharing a new project using CVS

Go To:

All Topics · Search · Bookmarks · Index

Eclipse Help Enhancements

- New specialized maps for modeling of Eclipse Help specifics like **contexts, indexes, cheat sheets** etc.
- **Import** of existing Eclipse Help Plug-ins
- **Tools for integration with development process** (Context IDs Management between development- and documentation teams)
  - Plug-in for Eclipse IDE
  - Wizards for ID Synchroniszation
- Support of dynamic **Context ID Mappings**
- Special **Validation** for Eclipse Help

Summary

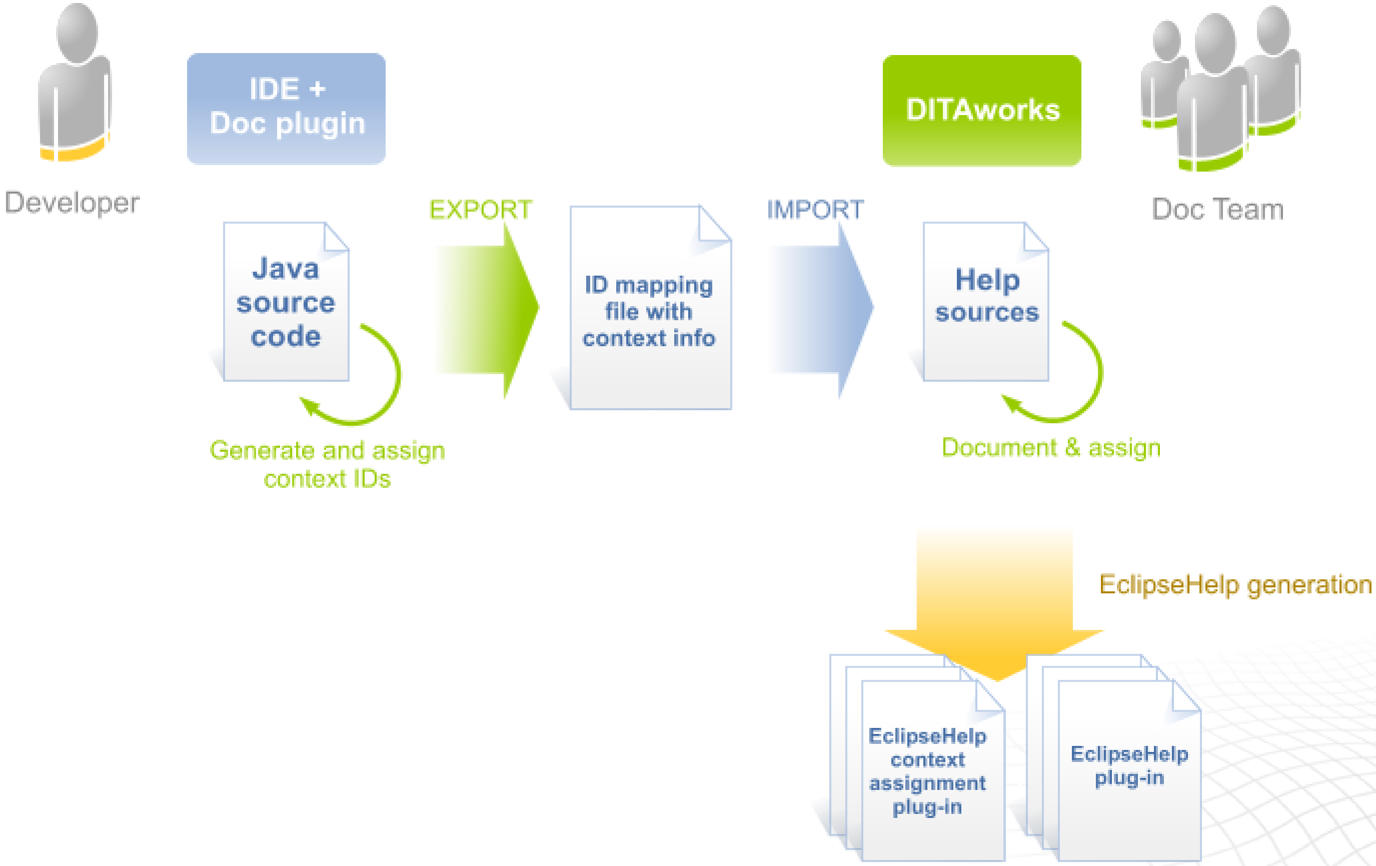
- ✓ DITAworks **addresses most of the challenges in the area of RCP application documentation** as a ready out-of-the-box product
- ✓ DITAworks **enables single-source approach** to the development of documentation under Eclipse
- ✓ DITAworks **can be easily integrated** with other Eclipse based tools
- ✓ DITAworks provides **IDE tooling to optimize collaboration**
- ✓ DITAworks pays special attention to **support of Eclipse help format**
- ✓ DITAworks is also a **good starting point for custom solutions** dealing with structured document generation

Demo

Demo details and goals

- ✓ Based on classical Eclipse “RCP Mail Template” example project
- ✓ Create a documentation for our sample RCP application in form of:
  - ✓ Eclipse Help
  - ✓ PDF
- ✓ Assign context help according to Dynamic Context Mapping Strategy
- ✓ Demonstrate the work environments for Dev and Doc teams
- ✓ Demonstrate the process

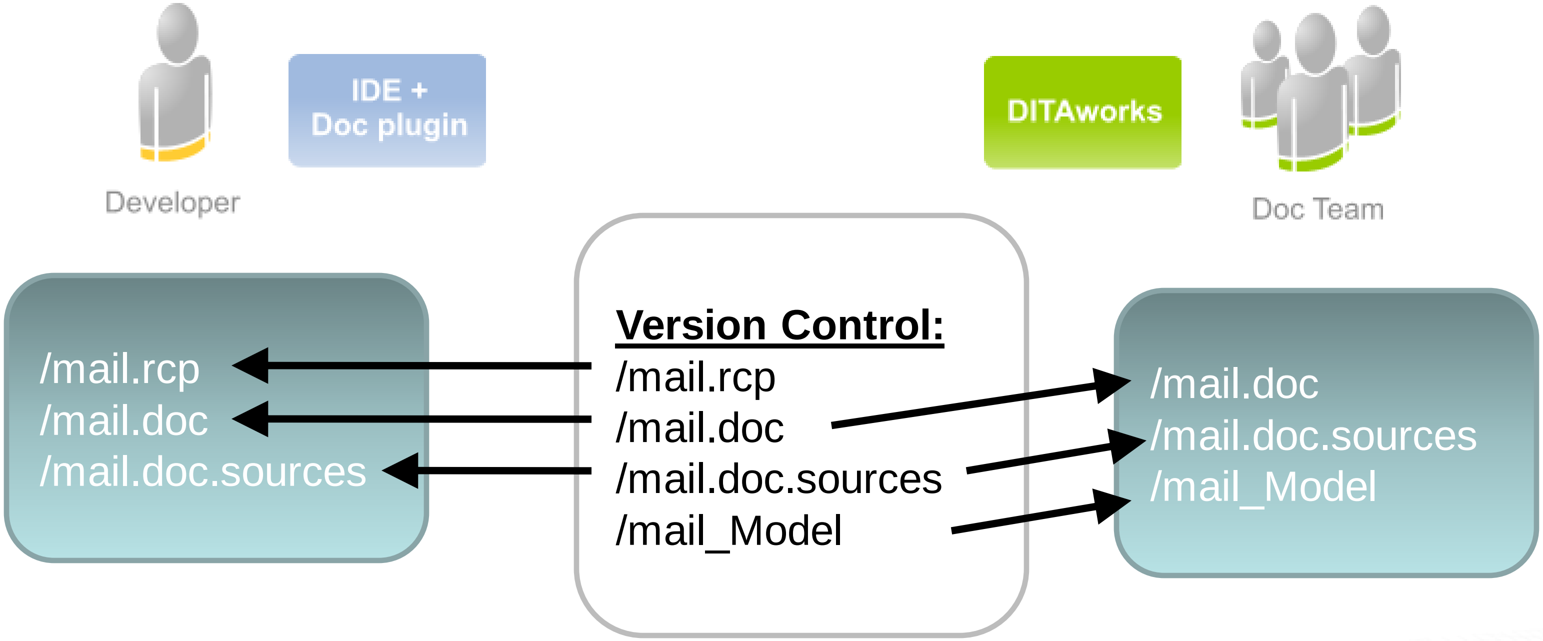
Eclipse Help Documentation Process



Step 1: setup infrastructure

- ✓ **2 workspaces** for Dev and Doc teams
- ✓ **Shared projects** (via VC) for content exchange
- ✓ **Different tooling:**
  - ✓ **Eclipse IDE + DITAworks IDE tools** for Dev
  - ✓ **DITAworks** for Doc teams

Work infrastructure



/mail.rcp
/mail.doc
/mail.doc.sources

Version Control:
/mail.rcp
/mail.doc
/mail.doc.sources
/mail_Model

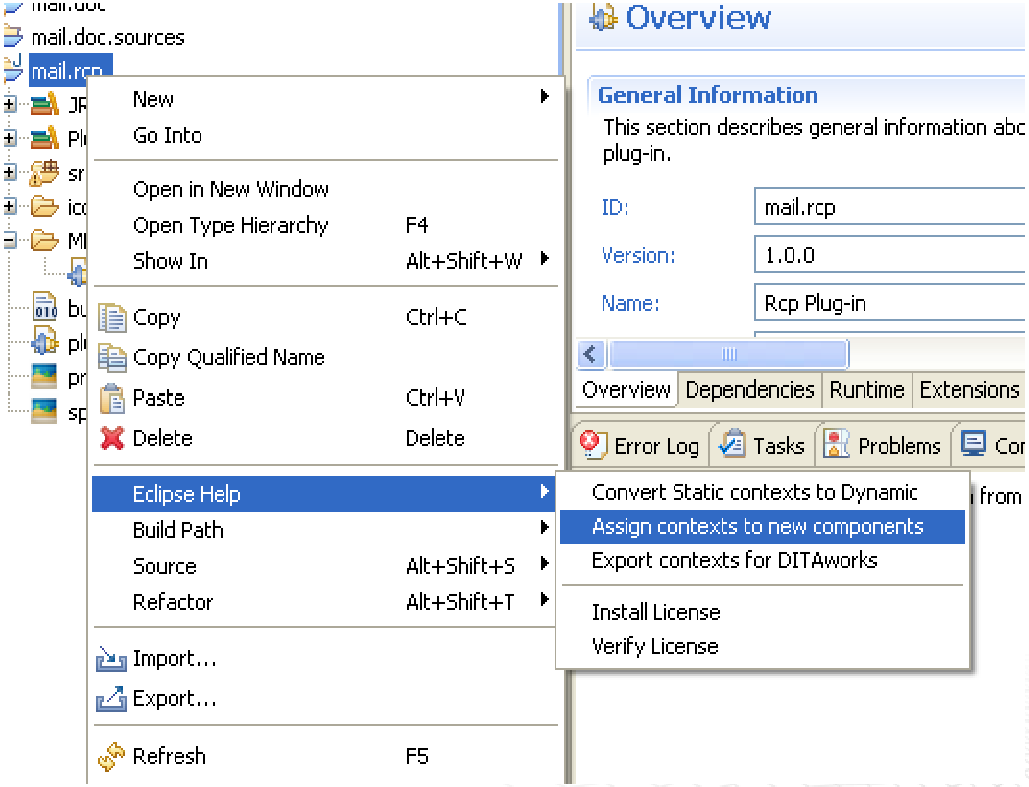
/mail.doc
/mail.doc.sources
/mail_Model

Step 2: Assign context IDs in code

Role: Developer

Tool: DITAworks IDE tooling

- ✓ **Find Java UI components** that require Help Context ID
- ✓ **Assign new Context IDs** using refactoring wizard.



This section describes general information abc plug-in.

ID: mail.rcp

Version: 1.0.0

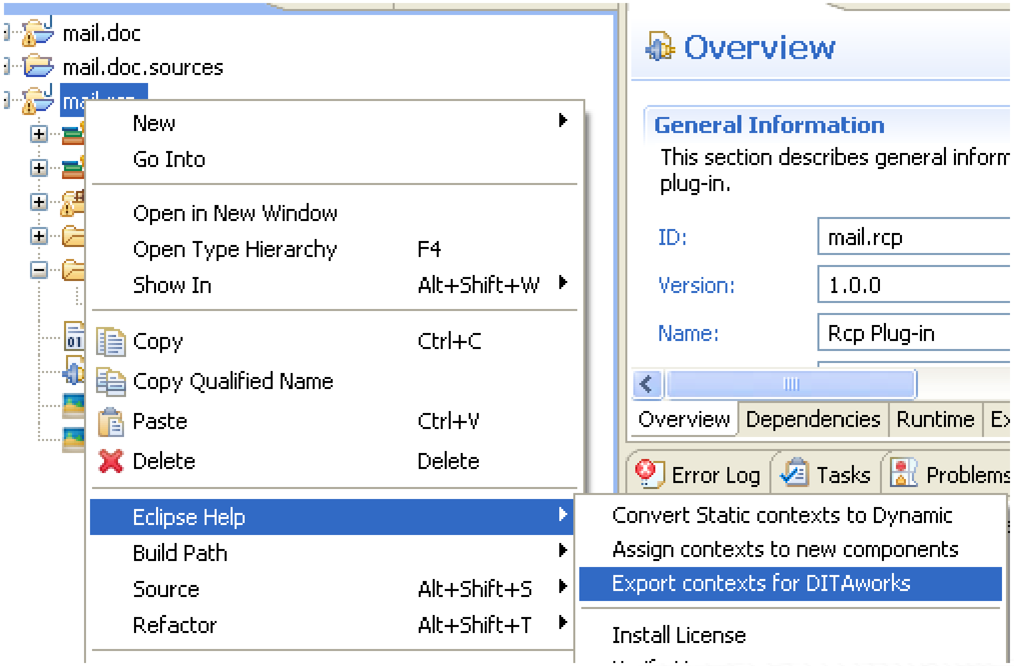
Name: Rcp Plug-in

Step 3: Export context IDs to shared project

Role: Developer

Tool: DITAworks IDE tooling

- ✓ **Run Export Wizard** for Context IDs
- ✓ **Describe** exported IDs (optional)
- ✓ **Store Context IDs** to shared help source project



This section describes general inform plug-in.

ID: mail.rcp

Version: 1.0.0

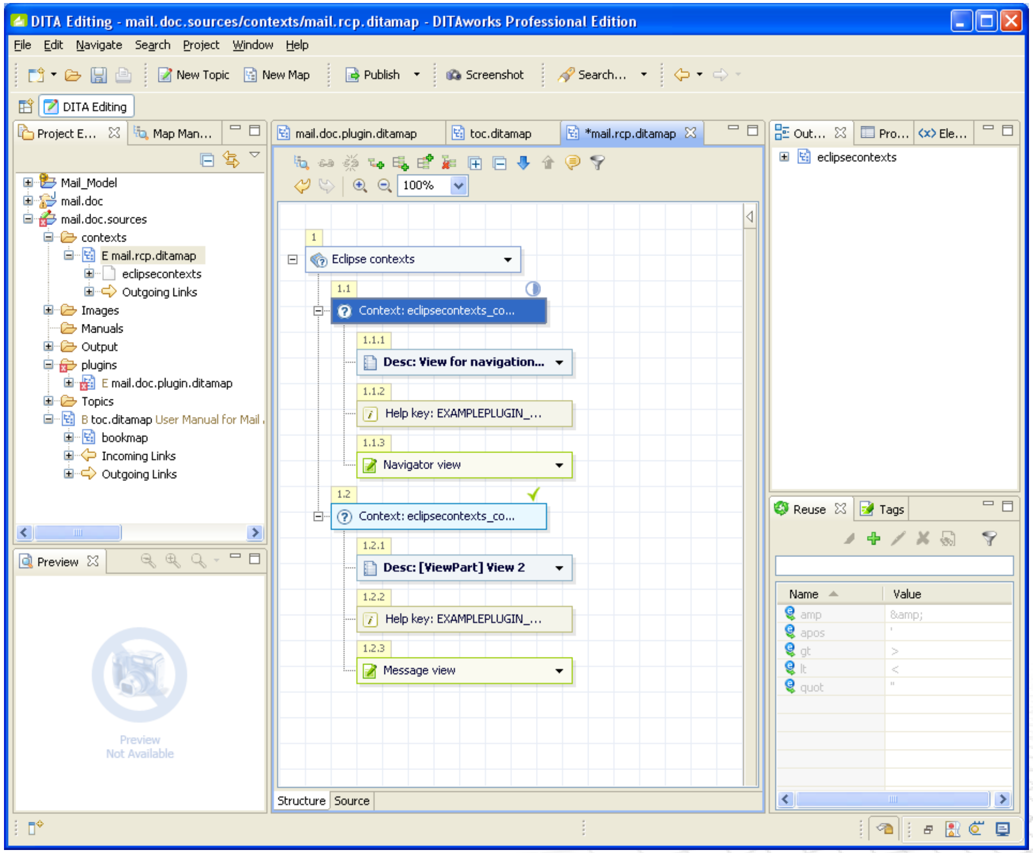
Name: Rcp Plug-in

Step 4: Assimilate context IDs and document

Role: Doc team

Tool: DITAworks

- ✓ **Open or Import** exported Context IDs
- ✓ **Document:** Assign existing topics. Write new topics.
- ✓ **Assign Context definitions** to help plugin

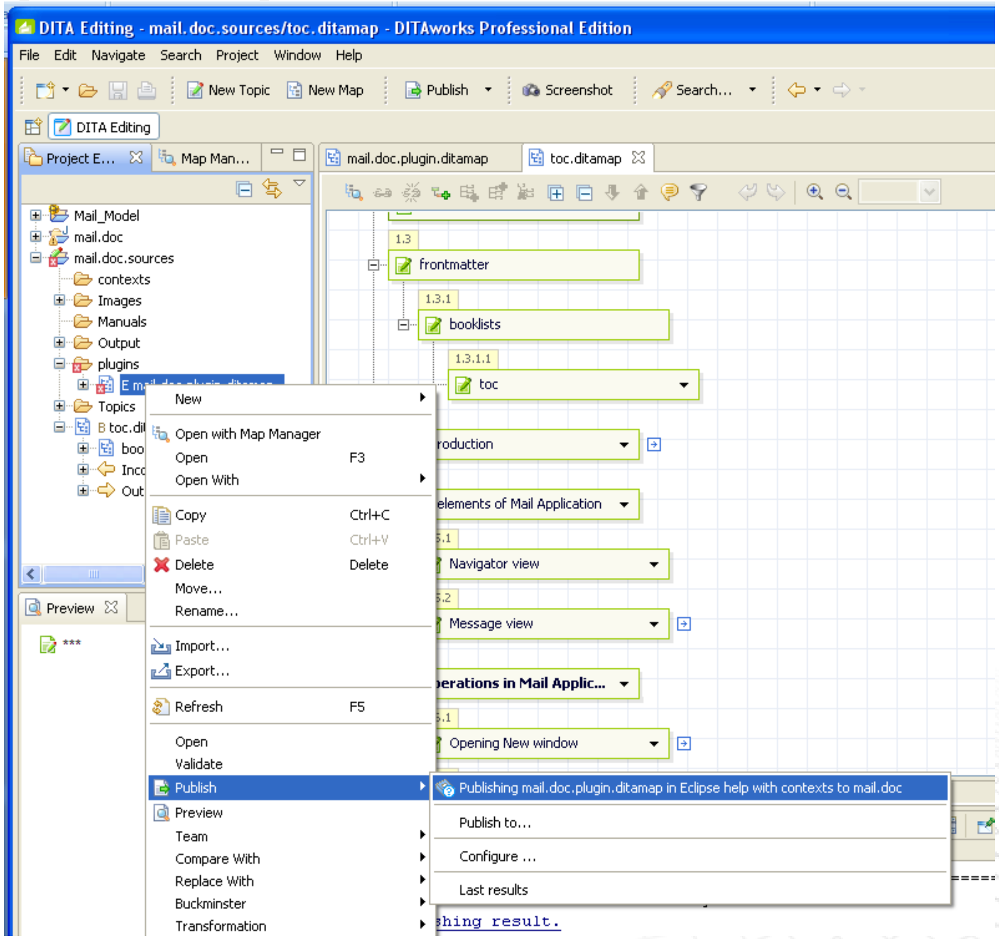


Step 5: Publish Eclipse help documentation

Role: Doc team

Tool: DITAworks

- ✓ **Setup publishing** configuration for Eclipse help plug-in
- ✓ **Run publishing process**
- ✓ **Share results.**

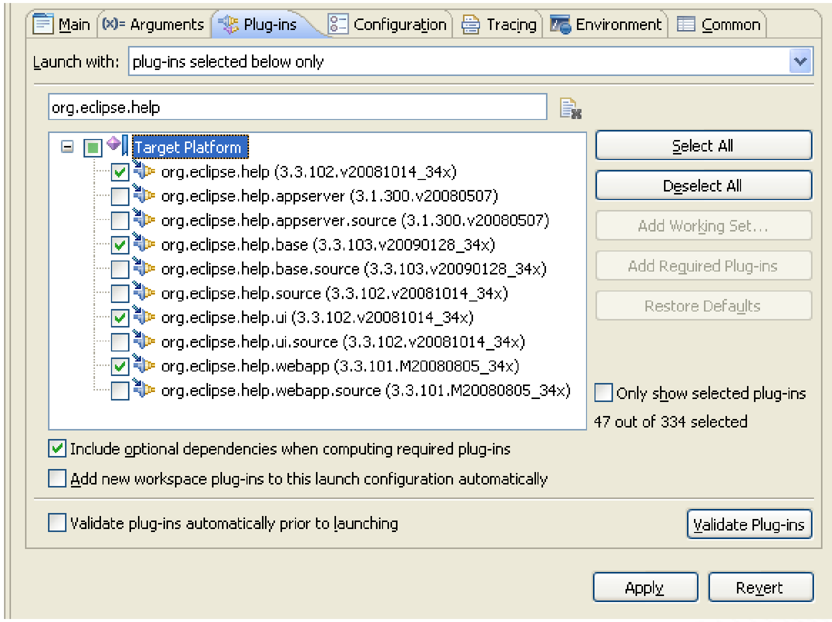


Step 6: Integrate help into application

Role: Developer

Tool: Eclipse IDE

- ✓ **Include Documentation bundles** into application.
- ✓ **Include** org.eclipse.help.* bundles
- ✓ **Build & Run**
- ✓ **Press F1** on one of the views in sample app



Launch with: plug-ins selected below only

org.eclipse.help

- ▣ Target Platform
  - ☑ org.eclipse.help (3.3.102.v20081014_34x)
  - □ org.eclipse.help.appserver (3.1.300.v20080507)
  - □ org.eclipse.help.appserver.source (3.1.300.v20080507)
  - ☑ org.eclipse.help.base (3.3.103.v20090128_34x)
  - □ org.eclipse.help.base.source (3.3.103.v20090128_34x)
  - □ org.eclipse.help.source (3.3.102.v20081014_34x)
  - ☑ org.eclipse.help.ui (3.3.102.v20081014_34x)
  - □ org.eclipse.help.ui.source (3.3.102.v20081014_34x)
  - ☑ org.eclipse.help.webapp (3.3.101.M20080805_34x)
  - □ org.eclipse.help.webapp.source (3.3.101.M20080805_34x)

□ Only show selected plug-ins

47 out of 334 selected

☑ Include optional dependencies when computing required plug-ins

□ Add new workspace plug-ins to this launch configuration automatically

□ Validate plug-ins automatically prior to launching

Step 7: Generate other formats

Role: Doc team

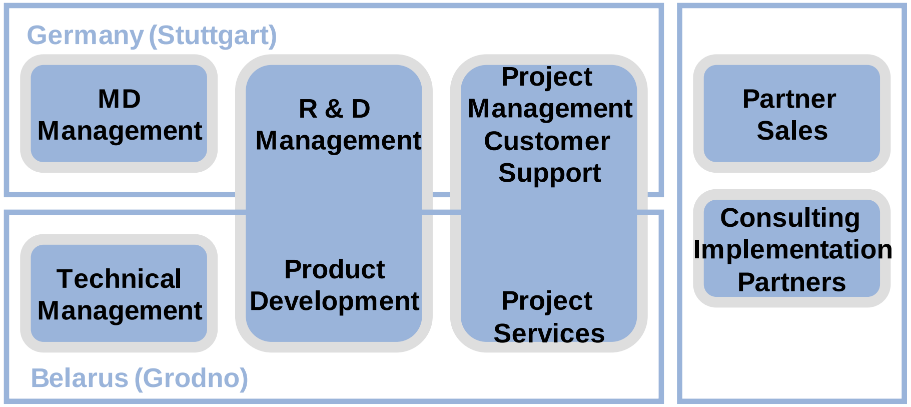
Tool: DITAworks

- ✓ **Setup publishing** configuration for other formats (f.e. PDF)
- ✓ **Run publishing process**
- ✓ **View results**

Q & A

About *instinctools

- *instinctools GmbH delivers Information Management solutions on Java technology since 2001, on Eclipse since 2007
  - Eclipse Application Design and Implementation Services
  - Eclipse Application Maintenance and Support
  - Tools for Technical Documentation (single source strategies)



- Management Team in Germany, Software Lab near shore (Belarus)
- Successfully serving premium customers like Daimler, Hubert Burda Media, Garant, EnBW and SMEs
- Proven management processes and reliable project delivery infrastructure
- Member of tekom, Eclipse Foundation

More Information

- Informationen about DITAworks
- Product videos on DITAworks Product Site
- Details on documenting Eclipse DITAworks Service Site and a Case Study on our Homepage
- You want to try it?

  Request a free of cost Evaluationlicense .

*instinctools™

New Ideas in Information Management

Sophienstr. 17

D-70178 Stuttgart

phone: +49 (0) 711 - 99 33 85 90
fax: +49 (0) 711 - 99 33 85 99
e-mail: **info@instinctools.com**
internet: **www.instinctools.de**
www.ditaworks.de