Berner Fachhochschule
Hochschule für Technik und Informatik
Software-Schule Schweiz

FLOW3

FLOW3 Development Environment for Eclipse

Requirements Specification
MAS-06-01.01

Abstract

Design and develop an Eclipse-based development environment for FLOW3 (an open-source PHP application framework).

Author

David Brühlmeier
Meisenweg 7
3186 Düdingen
Switzerland
david@bruehlmeier.com

Supervisor

Robert Lemke
Rathenaustrasse 23
23568 Lübeck
Germany
robert@typo3.org

Expert	Philippe Seewer
Class	MAS-IT 2006/02
Date	2008-04-02

Revision History

Revision 001	2008-03-18	David Brühlmeier
Version for review by supervisor Robert Lemke.		
Revision 002	2008-03-21	David Brühlmeier
Changes after feedback from supervisor Robert Lemke. This is the revision submitted to the Berner Fachhochschule.		
Revision 003	2008-03-27	David Brühlmeier
Changes according to feedback from expert Philippe Seewer after submission: Added information for all use-cases in which version they are planned for implementation and clarified that the creation of a new project is outside the scope of FLOW3DE (see preconditions in UC001).		
Revision 004	2008-04-02	David Brühlmeier
Changes according to feedback from expert Philippe Seewer after submission: Added planned release dates for the implementation of versions 0.1, 0.2 and 0.3.		

Table of Contents

Chapter 1. Introduction

1.1. Purpose

This document describes all requirements for the "FLOW3 Development Environment" (FLOW3DE). The requirements are documented as use-cases and as supplementary requirements (functional and non-functional).

1.2. Intended Audience

This document is intended mainly for the supervisor of this project, Robert Lemke, and the expert assigned for this Master Thesis, Philippe Seewer. However, anybody interested in this project is welcome to read it. You are invited to post comments on the TYPO3 5.0 mailing list (`TYPO3-project-5_0-general` on `lists.netfielders.de`).

1.3. Project Scope

In accordance with the proposal for the Master Thesis, the scope of this project is defined as follows:

- Complete description of a final verion of the FLOW3 Development Environment. This document represents this part.
- Development of a subset of the features. All the use-cases and requirements which will *not* be part of this first version of FLOW3DE are marked accordingly in this document.

1.4. Artifacts

The following artifacts shall be produced by the end of the project:

- Enterprise Architect Model

  Enterprise Architect is used as the CASE tool for this project. All use-cases, requirements and design information are documented in this tool. Upon build, this information is exported to a standardized XMI file and then converted to DocBook using XSL.

- Requirements Specification

  Parts of the requirements specification are generated from Enterprise Architect (such as the use-cases and the requirements), other parts are directly authored in DocBook (such as the introduction, etc.). The resulting files are merged using XLink and then rendered as PDF using XSL-FO.

- Master Thesis

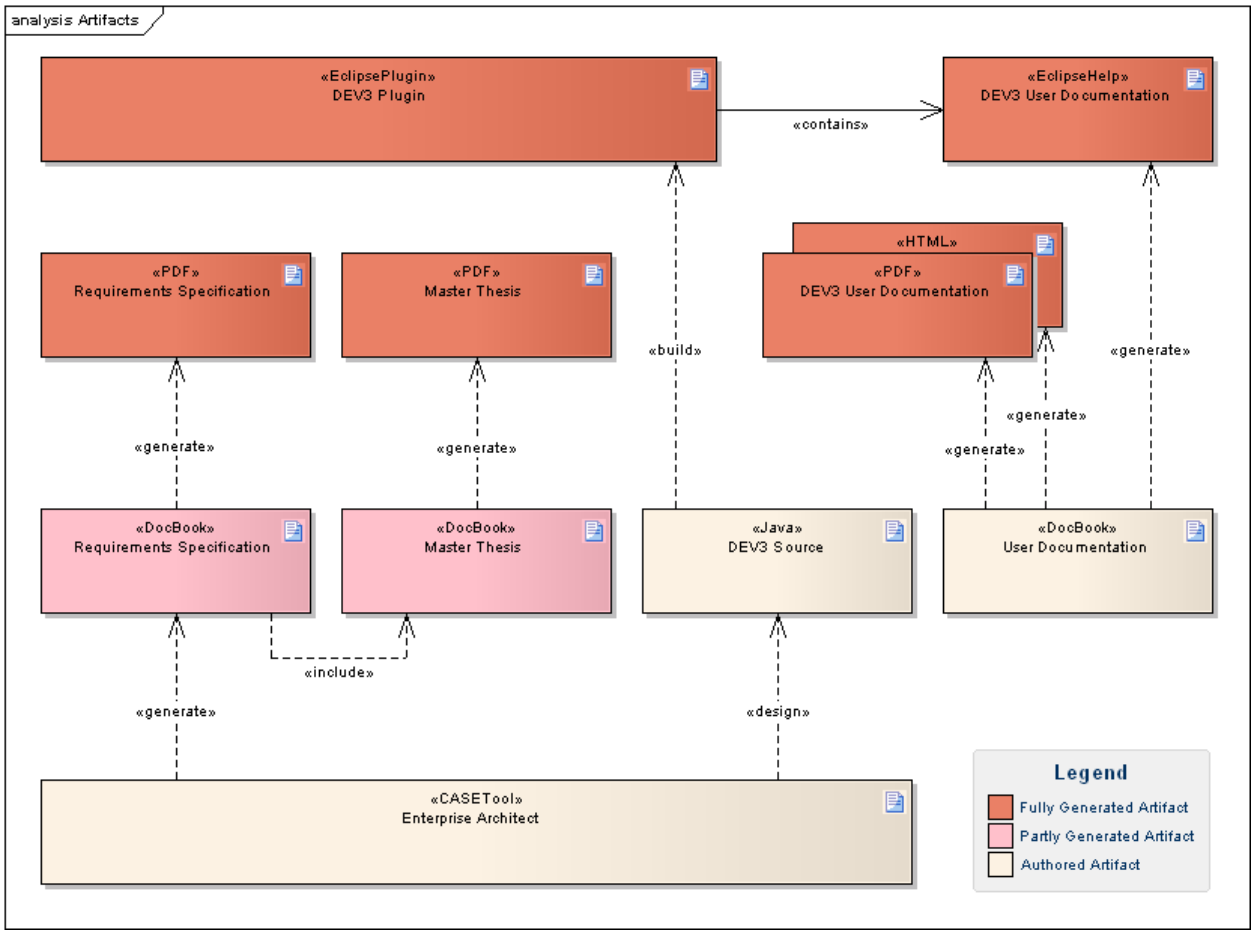
  Parts of the master thesis are generated from Enterprise Architect (such as the class diagrams), other parts are directly authored in DocBook (such as the introduction, design, etc.). The resulting files are merged using XLink and then rendered as PDF using XSL-FO.

- FLOW3DE Source

  The Java source code of the FLOW3DE is authored in Eclipse. Upon build, it is compiled as an installable Eclipse Plugin.

- User Documentation

  The user documentation is fully authored in DocBook. Upon build, it is rendered as PDF and HTML using XSL-FO and XSLT. The same source is also rendered as an Eclipse Help Plugin, also using XSLT, and then packaged with the Eclipse Plugin.



Artifacts to be produced by this Master Thesis

1.5. References

For additional information, please refer to the following online ressources:

- Project Site: `http://forge.typo3.org/projects/show/teams-flow3de`

  The project site contains information about the status of the project, such as development progress, issues, etc.

- Subversion Server: `https://svn.typo3.org/Teams/FLOW3DE`

  All results (code and documentation) is put under version control and can be downloaded from this subversion server.

- Wiki: `http://wiki.typo3.org/index.php/FLOW3DE`

  This wiki page is the central place to collect information to be discussed with the FLOW3 community.

- Eclipse Update Site: `http://typo3.org/go/downloads/flow3de`

  The packaged version of FLOW3DE can be downloaded from this update site using the standard Eclipse update mechanism.

Chapter 2. Overall Description

2.1. Product Perspective

FLOW3 is an application framework written in PHP. It stems from the TYPO3 community. The version 5.0 of TYPO3 is going to be a complete rewrite, based on a new architecture. The foundation of this new version is going to be FLOW3. For more detailled information, please refer to the FLOW3 Website: `http://flow3.typo3.org/`.

The FLOW3 Development Environment aims to be an integrated platform to develop applications based on FLOW3. It is an Eclipse Plugin and therefore integrates well with any existing development environments for Eclipse (such as PDE, the PHP Development Environment by Zend).

2.2. Product Features

FLOW3 brings some new concepts to the PHP world, such as Domain-Driven Design, Dependency Injection and Aspect-Oriented Programming. The FLOW3 Development Environment helps the developer to quickly use these new possibilities by offering the following features in its final version:

- Wizard to create new packages

  A wizard creates all folders and configuration files required by FLOW3. The developer can also choose from different templates to get him started faster.

- Configuration editor

  A configuration editor (like PDE in Eclipse) makes it easy to change configuration settings for a package, such as the `Package.xml` or `Components.php` file.

- Wizard to create new aspects

  A wizard creates scaffolding classes and configuration for new aspects.

- Editor for TypoScript 2.0

  A specialized editor helps the developer write TypoScript 2.0. The editor includes syntax highlighting and content assist, based upon the packages available.

- FLOW3 coding guidelines compliance

  All PHP code is checked for compliance with the FLOW3 coding guidelines as-you-type.

- Editor for the content repository

  A specialized editor helps editing the content repository, for both content and meta-data.

2.3. Operating Environment

The FLOW3 Development Environment is a plugin for Eclipse and will be developed in Java. As such, any operating system supporting Java can be used to run the FLOW3DE.

Other than a working Eclipse installations, there are no further requirements for running the FLOW3DE.

2.4. System Boundaries

In the first version, the FLOW3 Development Environment will only work locally, with no interaction from/to a FLOW3 server. In its final version, it shall be possible to connect the FLOW3DE with a target FLOW3 server, e.g. using the "Web Services Eclipse File System (WSEFS)" (refer to the glossary for details on WSEFS).

2.5. User Documentation

The user documentation will be provided in three different formats:

- Eclipse Help System

  The documentation integrated into Eclipse will offer context-sensitive help and general help using the menu "Help -> Help Contents".

- HTML

  The documentation will also be rendered as HTML for people who have not yet installed Eclipse or the FLOW3DE. This version shall be made available on `www.typo3.org`.

- PDF

  The documentation will also be rendered as PDF for people who prefer to download a printable version. This version shall also be made available on `www.typo3.org`.

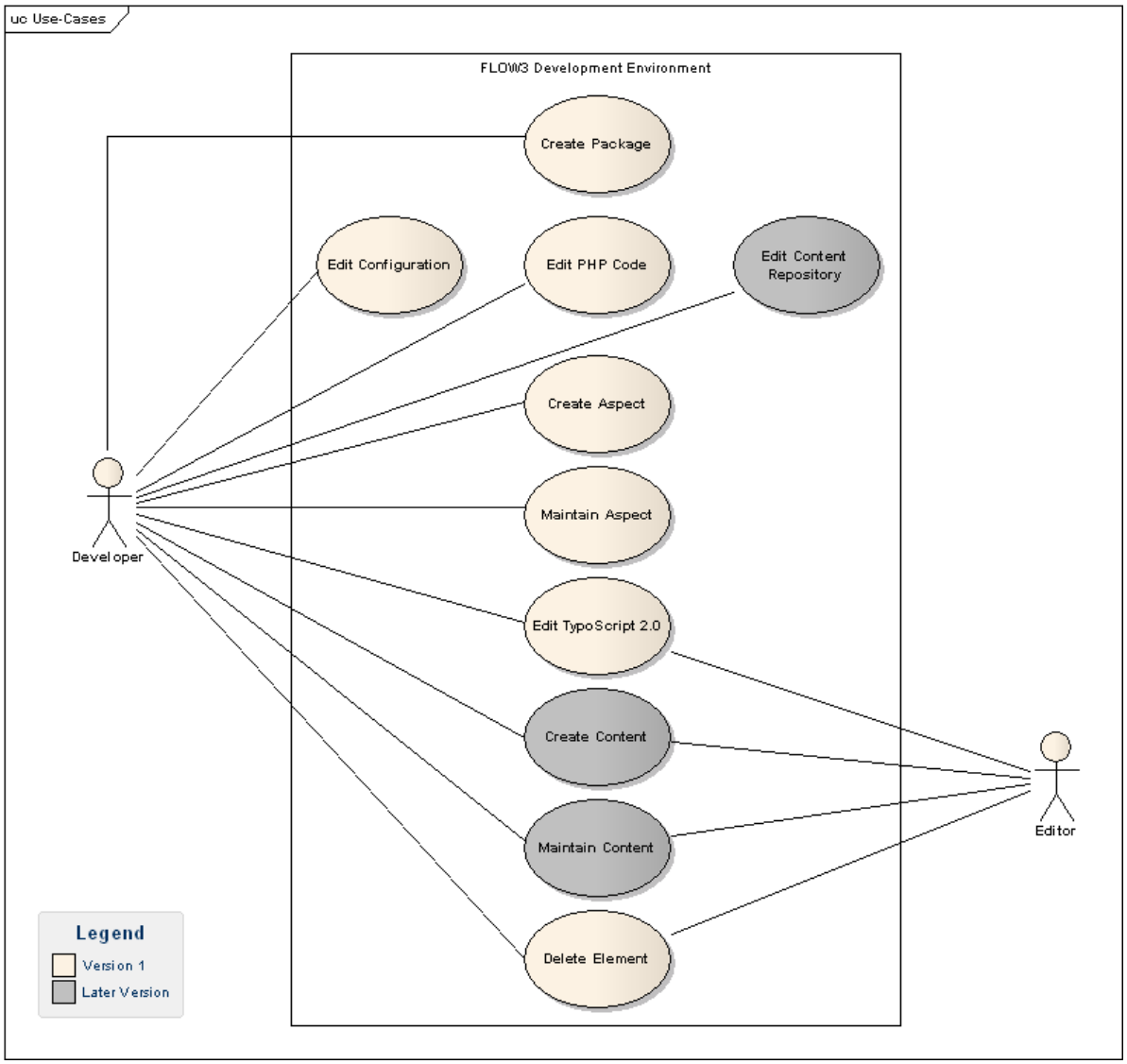
Chapter 3. Use-Cases

This chapter contains the detailled descriptions of all use-cases. The use-cases which will be developed during this Master Thesis are marked in the diagram below and in the descriptions. Each use-cases is assigend to a certain version number. The deadlines for the three versions planned are as follows:

- Version 0.1: 29.04.2008
- Version 0.2: 03.06.2008
- Version 0.3: 14.07.2008

The references from/to actors, use-cases and requirements are links and can be used to easily navigate within this document.

3.1. Overview



Use-Cases for the FLOW3 Development Environment in its final version

3.2. Create Package

ID	UC001
Version	This use-case is planned to be implemented in version 0.1 of the FLOW3DE.
Description	This use-case covers the creation of new packages.
Actors	Developer (AC001)
Preconditions	• A project must be available. Please note: The creation of a project is outside the scope of the FLOW3DE. Any type of Eclipse project can be used. Normally, it will be a PHP project created by Zend PDE, but it can be just a generic project or a project created by another PHP plugin.
Basic Path	1. Select folder in which the new package will be created. 2. Click on "Create new Package" wizard. 3. Fill in the package key (mandatory). 4. Fill in the package title (mandatory). 5. Fill in the package description (mandatory). 6. Fill in the version of the package (default = 0.0.1, mandatory) 7. Select a category (optional). 8. Fill in the name of the author (optional). 9. Fill in the e-mail address of the author (optional). 10. Select the additional folders which shall be created by the wizard. 11. Optionally choose the templates which shall be generated by the wizard. 12. Click on Finish.
Alternate Paths	• Wrong Or Missing Information 1. The user enters wrong information or doesn't enter a required information. 2. The system disables the "Finish" button and displays a message explaining how to solve the problem.
Postconditions	• All files and directories required by the FLOW3 Framework, such as the Package.xml file, are created • If the actor has selected templates, the chosen scaffolding classes are created.
Additional Requirements	• Custom Templates (RQ018) • Mandatory Directories/Files (RQ002) • Check For Valid Version Numbers (RQ006)

- Check For Valid Package Keys (RQ005)
- Templates (RQ004)
- New Package Wizard (RQ001)
- Default Values For New Package Wizard (RQ031)
- Optional Directories (RQ003)

3.3. Edit Configuration

ID	UC002
Version	This use-case is planned to be implemented in version 0.2 of the FLOW3DE.
Description	This use-case covers editing the meta information of a specific package.
Actors	Developer (AC001)
Preconditions	• The Package.xml file must be available.
Basic Path	1. The user double-clicks on the Package.xml file. 2. The system opens a specialized editor for this file. 3. The user changes the configuration as needed. 4. The user clicks on "Save".
Alternate Paths	---
Postconditions	• The Package.xml file is saved in the latest state. • The updated Package.xml is well-formed XML and is valid according to the appropriate schema.
Additional Requirements	• Configuration Editor GUI (RQ014) • Configuration Editor (RQ013)

3.4. Edit PHP Code

ID	UC003
Version	This use-case is planned to be implemented in version 0.2 of the FLOW3DE.
Description	This use-case covers editing PHP code within a package. Please note: The FLOW3DE will not provide a PHP editor on its own, but it will extend existing plugins, such as the freely available PDT (PHP Development Tools) plugin.
Actors	Developer (AC001)
Preconditions	• The PHP file to be edited must be available.
Basic Path	1. Double-click on the PHP file to be edited. 2. Make the changes to the PHP code. 3. Click on "Save".
Alternate Paths	---
Postconditions	• The PHP file must be saved in the latest state.
Additional Requirements	• Check For CGL-Compliance (RQ011) • AOP Annotations (RQ010) • New Exceptions (RQ012)

3.5. Edit Content Repository

This use-case will be part of a future release of the FLOW3DE. At the current state of development, it is still quite unsure what steps will be needed for this use-cases. For this reason, this use-case does not yet contain more specific requirements.

ID	UC004
Description	This use-case covers editing Content Repository (CR) definitions (the structure, not the content).
Actors	Developer (AC001)
Preconditions	• The content repository must be available (either locally or remotely).
Basic Path	---
Alternate Paths	---
Postconditions	• The changes to the content repository layout are saved consistently.
Additional Requirements	---

3.6. Create Aspect

ID	UC005
Version	This use-case is planned to be implemented in version 0.2 of the FLOW3DE.
Description	This use-case covers the creation of new Aspects.
Actors	Developer (AC001)
Preconditions	• The package for which the aspect shall be created must be available.
Basic Path	1. Choose the package for which the aspect shall be created. 2. Click on "Create new Aspect" wizard. 3. Enter the name of the aspect (mandatory) 4. Enter the description of the aspect (optional) 5. Enter the name of the author (optional) 6. Enter the e-mail address of the author (optional) 7. Choose to add pointcuts (optional) 8. Click on Finish.
Alternate Paths	• Wrong Or Missing Information 1. The user enters wrong information or doesn't enter a required information. 2. The system disables the "Finish" button and displays a message explaining how to solve the problem.
Postconditions	• The aspect is saved consistently.
Additional Requirements	• New Aspect Wizard (RQ008) • Check For Valid Aspect Names (RQ007) • Custom Templates (RQ018)

3.7. Maintain Aspect

ID	UC006
Version	This use-case is planned to be implemented in version 0.2 of the FLOW3DE.
Description	This use-case covers the whole lifecycle of an aspect, excluding its creation.
Actors	Developer (AC001)
Preconditions	• The aspect to be edited must be available.
Basic Path	1. Choose the aspect for which the advice shall be created. 2. Select "Create new Advice". 3. Enter the name of the advice (mandatory). 4. Define the advice type: "Before", "After returning", "After throwing", "After" or "Around" (mandatory). 5. Either refer to a named pointcut, define a new pointcut expression through selection or type in a custom pointcut expression. 6. Click on "Save".
Alternate Paths	• Create New "Class"-Pointcut 1. Choose the aspect for which the pointcut shall be created. 2. Select "Create new Pointcut". 3. Enter the name of the pointcut. 4. Choose the pointcut expression designator "Class". 5. Either directly enter the pointcut expression or choose a class name from a list. 6. Click on "Save". • Create New "Method"-Pointcut 1. Choose the aspect for which the pointcut shall be created. 2. Select "Create new Pointcut". 3. Enter the name of the pointcut. 4. Choose the pointcut expression designator "Method". 5. Optionally chose the visibility of the method ("public", "protected" or "private"). 6. Either directly enter the pointcut expression or choose a class name/ method name from a list. 7. Click on "Save".

- Create New "Within"-Pointcut

  1. Choose the aspect for which the pointcut shall be created.
  2. Select "Create new Pointcut".
  3. Enter the name of the pointcut.
  4. Choose the pointcut expression designator "Within".
  5. Either directly enter the pointcut expression or choose an interface/class name from a list.
  6. Click on "Save".

Postconditions

- The edited aspect is saved consistently, including any newly created advices or pointcuts.

Additional Requirements

3.8. Edit TypoScript 2.0

ID	UC007
Version	This use-case is planned to be implemented in version 0.3 of the FLOW3DE.
Description	This use-case covers editing TypoScript 2.0.
Actors	Editor (AC002), Developer (AC001)
Preconditions	• The package in which the TypoScript 2.0 definition is a part of must be available.
Basic Path	1. Select a package. 2. Double-click on the TypoScript 2.0 file. 3. The system opens a specialized TypoScript 2.0 editor. 4. Make all the changes needed. 5. Click "Save".
Alternate Paths	---
Postconditions	• The changed TypoScript 2.0 definitions are saved consistently.
Additional Requirements	• Syntax Highlighting (RQ017) • Content Assist (RQ016) • TypoScript 2.0 Editor (RQ015)

3.9. Create Content

This use-case will be part of a future release of the FLOW3DE. At the current state of development, it is still quite unsure what steps will be needed for this use-cases. For this reason, this use-case does not yet contain more specific requirements.

ID	UC008
Description	This use-case covers creating content (like a page, an article, etc) in the Content Repository.
Actors	Editor (AC002), Developer (AC001)
Preconditions	• The content repository must be available (either locally or remotely).
Basic Path	1. Choose the content repository. 2. Select "Create new content". 3. Choose the content type. 4. Enter the information based on the content type selected. 5. Click "Save".
Alternate Paths	---
Postconditions	• The newly created content is saved to the content repository.
Additional Requirements	• New Content Wizard (RQ009) • Custom Templates (RQ018)

3.10. Maintain Content

This use-case will be part of a future release of the FLOW3DE. At the current state of development, it is still quite unsure what steps will be needed for this use-cases. For this reason, this use-case does not yet contain more specific requirements.

ID	UC009
Description	This use-case covers creating, editing and deleting content in the Content Repository (CR).
Actors	Editor (AC002), Developer (AC001)
Preconditions	• The content repository must be available (either locally or remotely).
Basic Path	1. Select the content repository. 2. Select the content element to be edited. 3. Click "Edit". 4. Change the information in the content element. 5. Click "Save".
Alternate Paths	---
Postconditions	• The changed created content is saved to the content repository.
Additional Requirements	---

3.11. Delete Element

ID	UC010
Version	This use-case is planned to be implemented in version 0.3 of the FLOW3DE.
Description	This use-case covers the deletion of an element, such as a package, an aspect, a content element, etc.
Actors	Developer (AC001), Editor (AC002)
Preconditions	• The element to be deleted must be available.
Basic Path	1. Select the element to be deleted. 2. Choose "Delete". 3. The system asks for confirmation. 4. Click "OK".
Alternate Paths	• Abort Deletion 1. Select the element to be deleted. 2. Choose "Delete". 3. The system asks for confirmation. 4. Click "Cancel". 5. The system cancels the action without deleting the element.
Postconditions	• The element is deleted consistently from the system.
Additional Requirements	---

Chapter 4. Actors

4.1. User

ID AC003

Description This actor is used in the documentation to represent any actor. The other actors are derived from this actor.

Use-Cases ---

4.2. Developer

ID AC001

Description A developer is the main user of the system. The developer is interested in efficiently creating and maintaining FLOW3 packages and typically has good to expert skills in PHP. He might also maintain content, although this is not one of his typical tasks.

Use-Cases
- Edit Content Repository (UC004)
- Delete Element (UC010)
- Edit PHP Code (UC003)
- Create Aspect (UC005)
- Maintain Aspect (UC006)
- Edit Configuration (UC002)
- Create Package (UC001)
- Edit TypoScript 2.0 (UC007)
- Create Content (UC008)
- Maintain Content (UC009)

4.3. Editor

ID AC002

Description An editor is responsible for creating, editing and deleting content. He is not interested in the technical background of the system and typically is not a developer. His main focus is to efficiently use the system to communicate.

Use-Cases
- Edit TypoScript 2.0 (UC007)
- Maintain Content (UC009)
- Create Content (UC008)
- Delete Element (UC010)

Chapter 5. Additional Requirements

5.1. Functional Requirements

5.1.1. New Package Wizard

New Package Wizard

ID	RQ001
Description	A wizard must be provided to create a new package.
Realizing Use-Cases	• Create Package (UC001)

Mandatory Directories/Files

ID	RQ002
Description	The "New Package Wizard" must always create the directory "Meta" with the "Package.xml" file.
Realizing Use-Cases	• Create Package (UC001)

Optional Directories

ID	RQ003
Description	The "New Package Wizard" may optionally create the following directories: • Configuration • Documentation • Documentation/Manual/en_EN/ • Resources • Resources/Media • Resources/Templates • Resources/PHP • Resources/Java • Tests
Realizing Use-Cases	• Create Package (UC001)

Templates

ID	RQ004
Description	The "New Package Wizard" must optionally provide templates for various elements of a package, such as a Content Repository Design, TypoScript 2.0 snippets, etc.
Realizing Use-Cases	• Create Package (UC001)

Default Values For New Package Wizard

ID RQ031

Description For the following values, defaults must be configurable as project and/or as workspace defaults:

- Languages
- Author Name
- Author E-Mail
- Additional Folders

Realizing Use-Cases
- Create Package (UC001)

Check For Valid Package Keys

ID RQ005

Description The system must check if the package key entered by the user is valid. If the key is not valid, a warning must be issued, but the "Finish" button must not be disabled.

A valid package key must only use the characters a-z, A-Z and 0-9 and it must always start with an uppercase character.

Realizing Use-Cases
- Create Package (UC001)

Check For Valid Version Numbers

ID RQ006

Description The system must check if the version number entered by the user is valid. If the version number is not valid, a warning must be issued, but the "Finish" button must not be disabled.

Valid version numbers always contain three digits, separated by dots. No other characters are allowed. No negative digits are allowed.

Realizing Use-Cases
- Create Package (UC001)

5.1.2. New Aspect Wizard

Check For Valid Aspect Names

ID RQ007

Description The system must check if the aspect name entered by the user is valid. If the aspect name is not valid, the "Finish" button of the wizard must be disabled and the user must be notified.

Realizing Use-Cases
- Create Aspect (UC005)

New Aspect Wizard

ID RQ008

Description	A wizard must be provided to create a new aspect.
Realizing Use-Cases	• Create Aspect (UC005)

5.1.3. New Content Wizard

New Content Wizard

This requirement will be part of a future release of the FLOW3DE.

ID	RQ009
Description	A wizard must be provided to create new content elements.
Realizing Use-Cases	• Create Content (UC008)

5.1.4. PHP Editor

AOP Annotations

ID	RQ010
Description	The PHP Editor must support FLOW3-specific AOP annotations.
Realizing Use-Cases	• Edit PHP Code (UC003)

Check For CGL-Compliance

This requirement will be part of a future release of the FLOW3DE.

ID	RQ011
Description	The code entered in the PHP Editor must automatically be checked for compliance with the FLOW3 Coding Guidelines (CGL). Any problems must be marked in real-time and whenever possible a suggestion to correct the problem shall be available.
Realizing Use-Cases	• Edit PHP Code (UC003)

New Exceptions

ID	RQ012
Description	The PHP Editor must support the creation of new FLOW3 exceptions with a unique identifier (i.e. the current UNIX timestamp).
Realizing Use-Cases	• Edit PHP Code (UC003)

5.1.5. Configuration Editor

Configuration Editor

ID	RQ013

Description	An editor must be provided to make all configuration for a specific package. The editor must be able to read and edit all relevant package configuration files, such as the "Package.xml" file.
Realizing Use-Cases	• Edit Configuration (UC002)

Configuration Editor GUI

ID	RQ014
Description	The GUI of the Configuration Editor shall follow the patterns used in the PDE (Plugin Development Environment) of Eclipse.
Realizing Use-Cases	• Edit Configuration (UC002)

5.1.6. TypoScript 2.0 Editor

TypoScript 2.0 Editor

ID	RQ015
Description	An editor for TypoScript 2.0 must be provided.
Realizing Use-Cases	• Edit TypoScript 2.0 (UC007)

Content Assist

ID	RQ016
Description	The TypoScript 2.0 editor must provide content assist ("auto completion"). The content assist must recognize the currently installed packages and account for the TypoScript elements provided by these packages.
Realizing Use-Cases	• Edit TypoScript 2.0 (UC007)

Syntax Highlighting

ID	RQ017
Description	The TypoScript 2.0 editor must provide syntax highlighting.
Realizing Use-Cases	• Edit TypoScript 2.0 (UC007)

5.2. Non-Functional Requirements

5.2.1. Usability

Custom Templates

ID	RQ018
Description	It must be possible to provide custom templates for any of the wizards that offer templates.
Realizing Use-Cases	• Create Package (UC001) • Create Aspect (UC005)

- Create Content (UC008)

File-System Settings

This requirement will be part of a future release of the FLOW3DE.

ID	RQ019
Description	The settings of the local or remote file-system (and the connection properties for the remote file-system) must be made on project level. It shall be possible to define default settings on the workspace level.
Realizing Use-Cases	---

Local Or Remote File-System

This requirement will be part of a future release of the FLOW3DE.

ID	RQ020
Description	It must be possible to either work on the local or any remote file system.
Realizing Use-Cases	---

Multi-Language

ID	RQ021
Description	The system's GUI must support multiple languages.
Realizing Use-Cases	---

Online Documentation

ID	RQ022
Description	The system must provide an online documentation integrated into the Eclipse help system.
Realizing Use-Cases	---

Shortcuts

ID	RQ023
Description	All frequently used features of the system shall be accessible through keyboard shortcuts, without using the mouse.
Realizing Use-Cases	---

5.2.2. Reliability

Error Logs

ID	RQ024

Description	Upon system failures, the system shall produce error logs with all the debugging information needed to located and reproduce the error situation.
Realizing Use-Cases	---

Wrong Or Insufficient Input

ID	RQ025
Description	Upon wrong or insufficient user input, the system shall provide a concise error message including suggestions on how to solve the problem.
Realizing Use-Cases	---

5.2.3. Performance

Access Remote File System

ID	RQ026
Description	Accessing the remote file system shall take no longer than one second per request on average.
Realizing Use-Cases	---

Lazy Initialization

ID	RQ027
Description	The system must adhere to the general concept of "Lazy Initialization" of Eclipse to avoid long startup times.
Realizing Use-Cases	---

5.2.4. Supportability

Coding Guidelines

ID	RQ028
Description	The system shall strictly adhere to the Eclipse Coding Guidelines.
Realizing Use-Cases	---

Unit-Tests

ID	RQ029
Description	The system's main functionality must be tested by automated unit-tests. The code coverage shall be no less than 75 percent.
Realizing Use-Cases	---

Update Site

ID	RQ030

Description	The system must provide an update mechanism using an Eclipse Update Site.
Realizing Use-Cases	---

Appendix A. Glossary

Advice	An advice is the action taken by an aspect at a particular join point. Advices are implemented as methods of the aspect class. These methods are executed before and / or after the join point is reached.
Advice Chain	If more than one around advice exists for a join point, they are called in an onion-like advice chain: The first around advice probably executes some before-code, then calls the second around advice which calls the target method. The target method returns a result which can be modified by the second around advice, is returned to the first around advice which finally returns the result to the initiator of the method call. Any around advice may decide to proceed or break the chain and modify results if necessary.
After Advice	An after advice is executed after the target method has been called, no matter if an exception was thrown or not.
After Returning Advice	An after returning advice is executed after returning from the target method. The result of the target method invocation is available to the after returning advice, but it can't change it. If the target method throws an exception, the after returning advice is not executed.
After Throwing Advice	An after throwing advice is only executed if the target method throwed an exception. The after throwing advice may fetch the exception type from the join point object.
Around Advice	An around advice is wrapped around the execution of the target method. It may execute code before and after the invocation of the target method and may ultimately prevent the original method from being executed at all. An around advice is also responsible for calling other around advices at the same join point and returning either the original or a modified result for the target method.
Aspect	An aspect is the part of the application which cross-cuts the core concerns of multiple objects. In FLOW3, aspects are implemented as regular classes which are tagged by the @aspect annotation. The methods of an aspect class represent advices, the properties act as an anchor for introductions.
Aspect Oriented Programming (AOP)	Aspect-Oriented Programming (AOP) is a programming paradigm which complements Object-Oriented Programming (OOP) by separating concerns of a software application to improve modularization. The separation of concerns (SoC) aims for making a software easier to maintain by grouping features and behaviour into manageable parts which all have a specific purpose and business to take care of.
Before Advice	A before advice is executed before the target method is being called, but cannot prevent the target method from being executed.
CGL	CGL stands for "Coding Guidelines". The FLOW3 CGL are available online: http://5-0.dev.typo3.org/guide/bk03pt05ch05.html There is also a package which checks FLOW3 code for CGL compliance. It is written in PHP: http://forge.typo3.org/projects/show/packages-flow3cgl
Content	In the context of a Content Management System such as TYPO3, "content" refers to any element which is visible on the website, such as an article, a news item or a calendar entry.

Content Repository (CR)	The Content Repository (CR) is the blackbox where records (pages, news items) and files are stored. From outside it looks like a kind of object database - internally it's based on relational databases like MySQL and the file system.
FLOW3	FLOW3 is an application framework written in PHP. It stems from the TYPO3 community. The version 5.0 of TYPO3 is going to be a complete rewrite, based on a new architecture. The foundation of this new version is going to be FLOW3.
FLOW3 Development Environment (FLOW3DE)	Eclipse plugin for the development and maintenance of FLOW3 packages.
Introduction	An introduction redeclares the target class to implement an additional interface. By declaring an introduction it is possible to introduce new interfaces and an implementation of the required methods without touching the code of the original class.
Join Point	A join point is a point in the flow of a program. Examples are the execution of a method or the throw of an exception. In FLOW3, join points are represented by the T3_FLOW3_AOPJoinPoint object which contains more information about the circumstances like name of the called method, the passed arguments or type of the exception thrown. A join point is an event which occurs during the program flow, not a definition which defines that point.
Package	FLOW3 is a package-based system. Packages act as a container for many different purposes: Most of them contain PHP code which adds certain functionality, others only contain documentation and yet other packages consist of templates, images or other resources.
Pointcut	The pointcut defines a set of joinpoints which need to be matched before running an advice. The pointcut is configured by a pointcut expression which defines when and where an advice should be executed. FLOW3 uses methods in an aspect class as anchors for pointcut declarations.
Pointcut expression	A poincut expression is the condition under which a joinpoint should match. It may, for example, define that joinpoints only match on the execution of a (target-) method with a certain name. Pointcut expressions are used in pointcut- and advice declarations.
Target	A class or method being adviced by one or more aspects is referred to as a target class /-method.
TypoScript	TypoScript is a configuration language of TYPO3. In version 2.0, TypoScript has become fully object oriented and is now part of FLOW3.
Web Services Eclipse File System (WSEFS)	The WSEFS provides an abstracted client-server based file system integrated into Eclipse. It was developed as a master thesis by Andreas Tschirpke at the University of Liverpool.