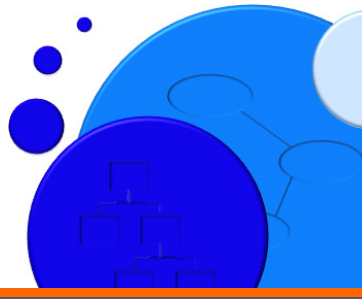
Cradle-7
From concept to creation...

3SL

Traceable and Tracked Document Management from Cradle

RA001/03 March 2017



© March 2017 3SL. All rights reserved.
Cradle is a registered trademark of 3SL in the UK and other countries. All rights reserved.
All other trademarks are the property of their respective owners.

http://www.threesl.com
salesdetails@threesl.com
support@threesl.com

Structured Software Systems Ltd (3SL)
Suite 2, 22a Duke Street
Barrow-in-Furness
Cumbria LA14 1HH, UK
Tel: +44 (0) 1229 838867
Fax: +44 (0) 1229 870096
Regd: 2153654 VAT: GB 473 2757 28

Contents

Copyright © March 2017 Structured Software Systems Ltd

Cradle is a registered trademark of Structured Software Systems Limited. All other products or services in this document are identified by the trademarks or service marks of their respective organisations.

Introduction

Virtually all projects use documents, as sources of information or as outputs to customers or to other groups. This include projects using agile processes. It is common for some documents to have a special significance, for example to:

- Define the customer's technical and commercial requirements
- Define the acceptance criteria
- Record the system design
- Record the verification or validation of some or all of the system

While these documents may be produced many times, specific instances of such documents often have a special significance, shown by one or more of:

- A particular issue number
- Stamping with a marking
- Bearing a seal
- One or more signatures on printed copies

The legal or commercial significance of documents forces most projects to have a document-centric process. Irrespective of the extent to which the data-centric approaches of requirements management and systems engineering have been adopted by a project, the project is typically managed by the issue states of its key input and output documents.

Subject

This paper considers how to reconcile the document-based demands of the customer and the project's own organisation with the data-centric perspective that is natural for requirements management and systems engineering techniques.

In particular, how can the problems traditionally associated with documents be reconciled with the needs for formal traceability and tracking inherent in formal systems engineering processes.

The Needs 1

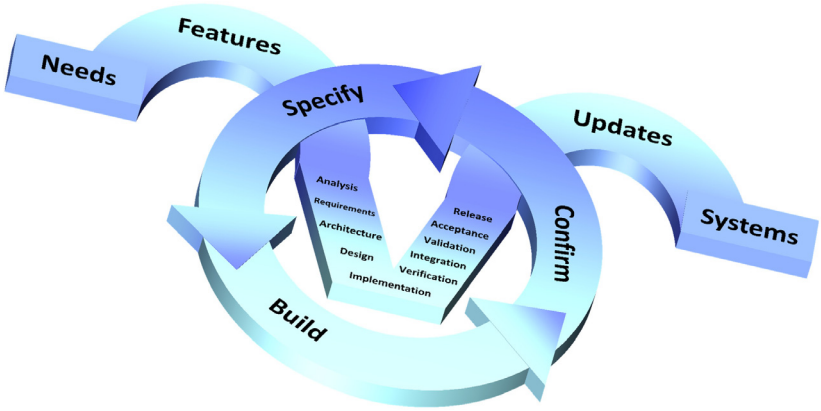
The process in most projects is based on each user story, requirement, use case, verification and so on being an individual item that can be linked to other items and managed. However, the relationship with a customer is invariably based on documents. This creates fundamental needs, to be able to:

- Report which items have been created or updated from a specific instance of an input document
- List the differences between the items updated from two input documents or from two instances of a single input document
- List which documents have created or updated specific items
- Report which items have been published in a given instance of an output document
- List the differences between the items that have been published in two output documents or two instances of a single document
- List all documents in which specific items have been published

For example, consider the following typical customer questions:

- ***Which versions of which requirements are in this SRD?***
- ***Which requirements have changed between these two SRD issues?***
- ***What are the differences between the designs in the original and current issues of the SDD?***
- ***Requirement 1.2.3.4 needs to change, so which documents need to be re-issued?***

It may be imagined that agile projects do not have the same need to manage input documents as projects using more traditional phase-based processes. Agile projects do, or should, have the same need. Most agile processes are based on a ***feature backlog*** expressed as ***user stories*** (the terms vary), sometimes grouped into ***epics*** (again, the term used may vary between projects).



But user stories are not requirements. At best, they are convenient, short, summaries of one or more user needs. They contain too little information to be validated by anything other than the most general user experience functional tests. Hence each user story will, or should, link to a collection of user requirements that are traced back to the users' original needs in the input documents.

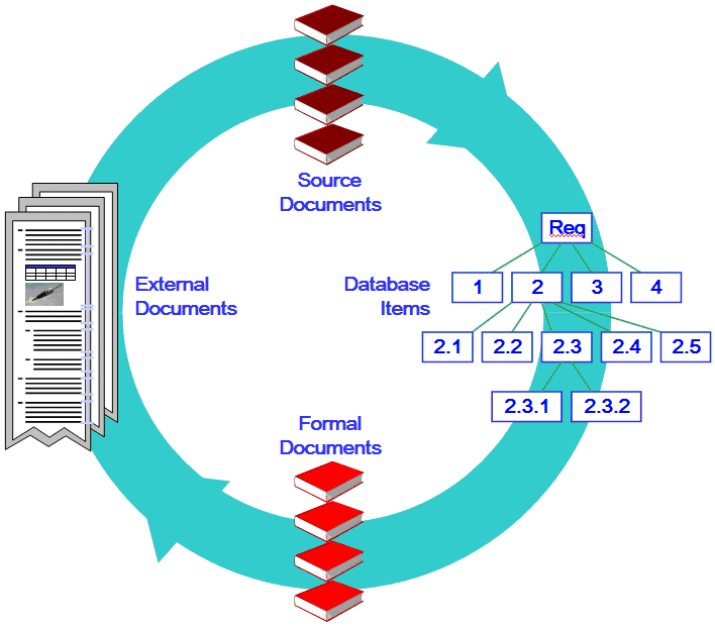
The essence of agile is short-cycle times, variously called ***sprints*** or ***iterations***, and the idea that a ***viable system*** must exist at the end of ***every*** iteration. Nonetheless, each iteration is a full V-lifecycle, with analysis, design, implementation and test activities.

Each iteration requires traceability, coverage and documentation. The short timescales create an urgent need for automated documentation.

The Solution 2

Cradle meets the needs with *source documents* and *formal documents*:

- Source documents are documents loaded into the database to create or update items, such as user requirements, acronyms, standards, references or acceptance criteria
- Formal documents are documents published from the database to fulfil a specific need of an external group, typically a customer



Cradle provides facilities to process and correlate the documents and database, and process them cyclically, if needed.

This allows a project to:

1. Work in a data-centric manner:
   - Where all data is stored once, but used many times
   - Where the database is a *single point of truth*
2. Interface fully and seamlessly with its document-centric environment
3. Track all changes to every input document through the database
4. Track every piece of every output document
5. Publish any number of documents that are guaranteed to be complete and consistent

The Benefits 3

Cradle allows a project to avoid the problems traditionally associated with documentation produced manually:

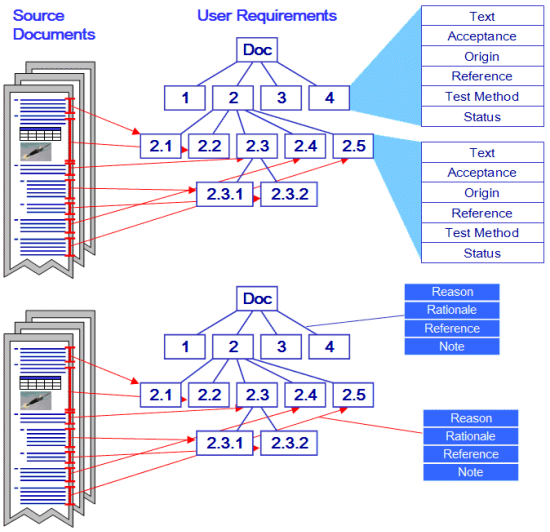
- ***Incompleteness***: the documents cannot be guaranteed to be complete, and attempts to ensure completeness are time consuming and therefore costly
- ***Inconsistency***: the same information appears in several documents and attempts to ensure consistency between these occurrences are time consuming and costly
- ***Obsolescence***: the underlying information changes rapidly and the time taken to manually prepare, quality assure and cross check documents means they become obsolete soon after release
- ***Irrelevance***: the obsolescence of manually produced documents makes them irrelevant, causes them to be ignored and to lose the authority intended by their authors

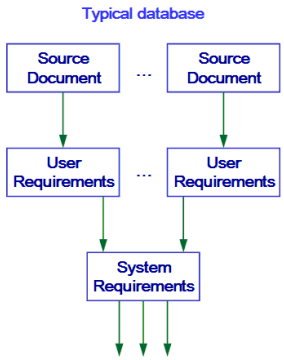
Source Documents 4

Any number of source documents containing requirements, standards, regulations or other inputs can be loaded into a Cradle database using the **Document Loader** tool, that:

- Creates a source document in the database
- Copies the external document into the source document, thereby protecting it from change

- Divides the source document into ***source statements***, one for each section, paragraph, table (row, cell or table) being loaded
- Creates one item in the database for each source statement
- Cross references the items into a hierarchy corresponding exactly to the hierarchy of sections and subsections in the document
- Cross references each item to its originating source statement, so linking each item to its origin text in the source document

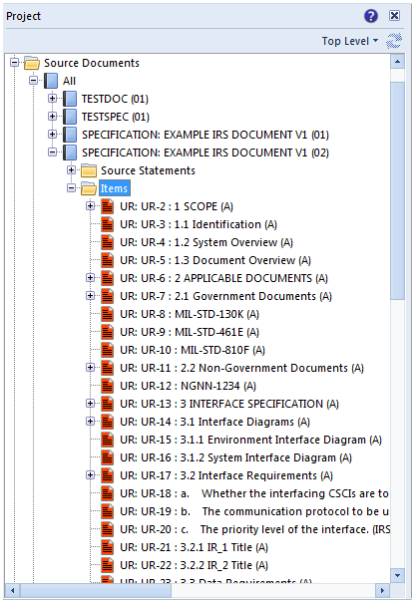




Cradle keeps a detailed record of the database items that are associated with each version of each source document.

When a new version of a source document is received, **Document Loader** highlights the differences between it and the latest version in the database. This allows the project to decide if the new document is to be loaded.

If a new source document version is loaded, Cradle repeats the above process, updating database items, creating a new set of cross references, and recording the instances of each database item that correspond to the new version of the source document.



Impact analyses from these changes allow an assessment to be made of the acceptability of new source document versions. Unacceptable new versions can be deleted pending discussion with their authors, causing Cradle to revert to the previous versions of such source documents.

Source document management therefore means:

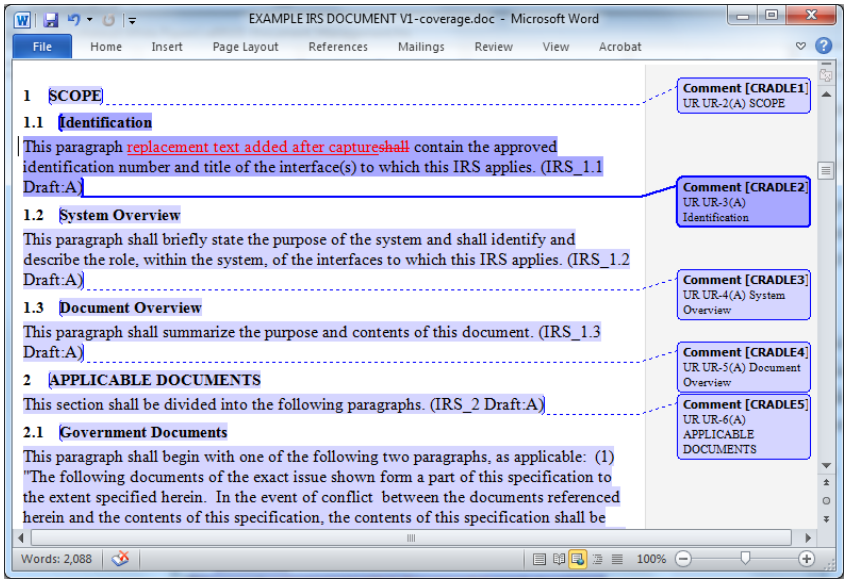
1. Any number of source documents can be captured
2. Each source document can exist in any number of versions
3. Each source document is cross referenced to database items created, or updated, by it
4. Detailed records are maintained of which instances of items are associated with each version of each source document:
   - That can be viewed from items to documents
   - That can be viewed from documents to items
5. Source document versions can be compared
6. The sets of database items associated with source documents' versions can be compared
7. Source document versions can be deleted, which:
   - Reverses all changes to database items
   - Deletes the source document version and the cached document
   - Reverts to the previous version of that source document

- Reverts to the set of cross references associated with the previous version of that source document

Collectively, these provide a comprehensive facility allowing all aspects of all source documents, and their updates, to be tracked, managed and related to the rest of the project's information.

Customer Compliance 5

Many projects operate within strict regulatory frameworks in which it is vital to demonstrate full compliance with customer documents. The source document mechanism also meets this need by providing three types of ***coverage analysis*** of any source document:



1. Which parts of the source document have, and have not, been captured into the database
2. Where is each part of the source document stored in the database, shown as a comment
3. All differences between the original text and the text stored in the database, shown in red using strikethrough and underline markup

This facility allows you to prove:

- Which part(s) of documents have been captured
- Where each document part is stored in the database
- That the customer's document's content has not been changed once loaded into the database, or has only been changed in ways agreed by the customer

Since all further compliance reports are based on the database items that contain the customer's information, these document compliance views are the vital first step, by confirming the validity of these database items.

Formal Documents 6

Most projects produce their significant, formal, output in documents. Typical examples are:

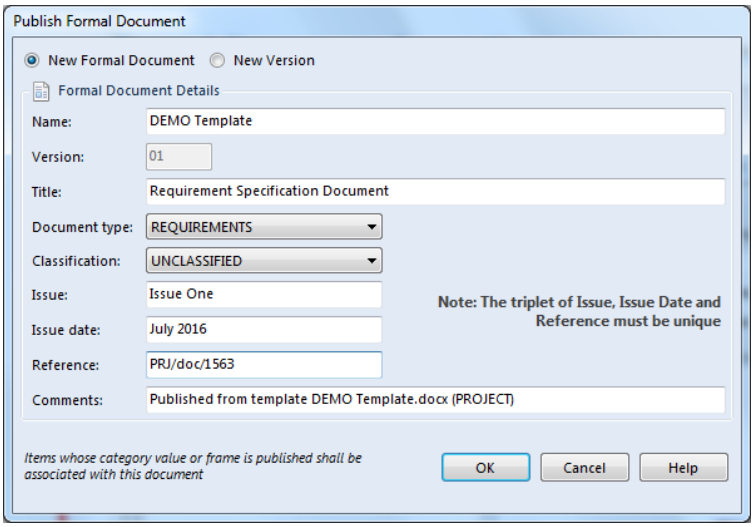
- Concept of Operations (CONOPS)
- System Requirements Document (SRD)
- System Design Document (SDD)
- Subsystem Design Documents (SSDDs)
- Interface Specifications (IFSs)

Such documents are often associated with major project milestones

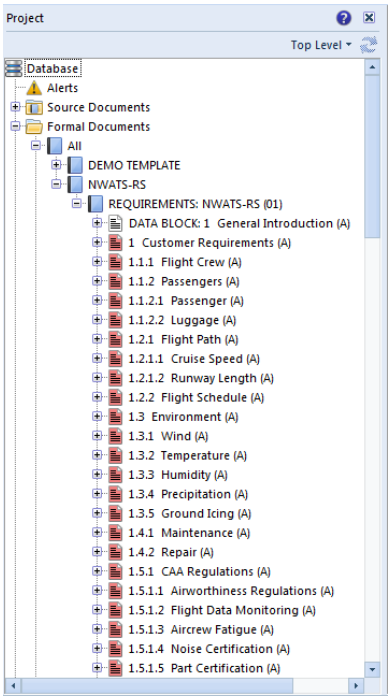
and sometimes with stage payments.

The **Document Publisher** tool is used to generate documents from a database. The form, structure and content of a document is described in a *template* which is itself a document that contains references into the database to retrieve the information to be published. One template is created for each style of document that your project will produce.

Each template can be used many times to publish items into many instances of a document, but when a significant project deliverable is to be produced, this same template is published in a way that marks the resulting document as a *formal document*. That is:



- Formal documents are published using the same templates as ordinary documents
- Users decide if a document is to be published as a draft, or as a formal document
- The content of a formal document is intrinsically no different to any other document produced from the same template
- Users specify a *title*, *issue*, *issue date* and *reference* when a formal document is published, which can be printed anywhere in the formal document, typically on the cover page.
- Cradle does further processing to publish a formal document



It is normal for formal documents to be published from the latest baseline since this information has been reviewed and cannot change. However, Cradle does not force this to be so.

Cradle keeps a detailed record of the database items that have been published in each version of all formal documents.

Users can browse which items have been published in each formal document, as reports and interactively through the Cradle UIs.

As the information in the database evolves, new versions of existing formal documents can be published as new issues of these documents.

Formal document management therefore means:

1. Any number of formal documents can be published
2. Each formal document can exist in any number of versions
3. Cradle maintains detailed records of which items are associated with each version of each formal document
   - That can be viewed from items to documents
   - That can be viewed from documents to items
4. Formal document versions can be compared
5. The sets of database items associated with formal document versions can be compared

6. Formal document versions can be deleted, which:
   - Deletes the formal document and the cached document
   - Deletes the association between that version of the formal document and the items published in it

Collectively, these provide a comprehensive facility allowing all aspects of all formal documents, and their updates, to be tracked, managed and related to the rest of the project's information.

Summary 7

When tracking input source documents, Cradle:

- Keeps a record of the external documents that have been loaded
- Keeps a record of which items were loaded from each version of each source document

When publishing to formal documents, Cradle:

- Keeps a record of the formal documents that have been published
- Keeps a record of which items were published in each version of each formal document

Collectively, these provide full traceability from every issue of all input documents, into the database, to every issue of all output document and full tracking of the contents of all input and output documents,