Writing a book with RUP and IBM Rational RequisitePro

by **Christophe Tournier**
Technical Representative
Rational Software
IBM Software Group
Lausanne, Switzerland

People often ask whether it's possible to use IBM Rational Unified Process,® or RUP,® for nonsoftware projects. Well, I've tried it: I used a lightweight version of RUP when I wrote my first book, a handbook of Moroccan cooking.

Writing a book is in many ways similar to developing a software project. As a writer, you need a vision and a "story." You must define the overall structure of the book and clearly express all the ideas in a consistent manner. But writing involves invention and discovery as well as expressing what you already know. After creating a plan, you must control the process, constantly revising and improving successive drafts, but staying with the broad objectives you have set for the book. And at the end of the creative process, you need an editor to polish the text and a publisher to distribute the book to readers. The book is the final project artifact for a writing project, just as the executable code is the main artifact of a software project.

This article describes the process I followed in writing my book. It shows how I used IBM Rational tools, provides examples of the artifacts I created to achieve my goals, and discusses my results. In addition to RUP and IBM Rational RequisitePro,® I used IBM Rational SoDA®, and IBM Rational ClearCase.® RUP served as a process framework, providing best practices and guidance through all the activities I performed. The other IBM Rational tools helped me automate and simplify many of these activities. In particular, IBM Rational RequisitePro's tight integration with Microsoft Word gave me fast access to information stored within a standard database. IBM Rational SoDA helped me extract data to create customized Word documents, and IBM Rational ClearCase helped me track the

different versions of my documents.[1]

Planning and implementing the project

I began my writing project by prioritizing my main objectives:

- Write a practical book offering a broad overview of the topic for beginners and practitioners.
- Use IBM Rational tools to save time and keep me organized.
- Explore RUP in the context of a nonsoftware project.

Before writing anything, I identified four major activities for setting up my own simple writing process:

- *Activity 1*: Map the writing project to a generic RUP software development project.
- *Activity 2*: Produce a Development Case.
- *Activity 3*: Develop a Requirements Management Plan.
- *Activity 4*: Identify and mitigate risks.

I discuss each of these activities in detail in the following subsections.

Activity 1: Map the writing project to a generic RUP software development project

My mapping between a software process and the writing process focused on three points of view:

- **Dynamic view**: a description of the four development phases I would use.
- **Static view**: a description of the analogy between RUP disciplines and writing disciplines.
- **Best practices and tools view**: a description of the best practices and tools I would use to create artifacts.

The following subsections discuss these points of view. Figure 1 shows the disciplines and phases in the standard version of RUP. Figure 2 maps RUP to a writing project, showing the static and dynamic aspects of this mapping.

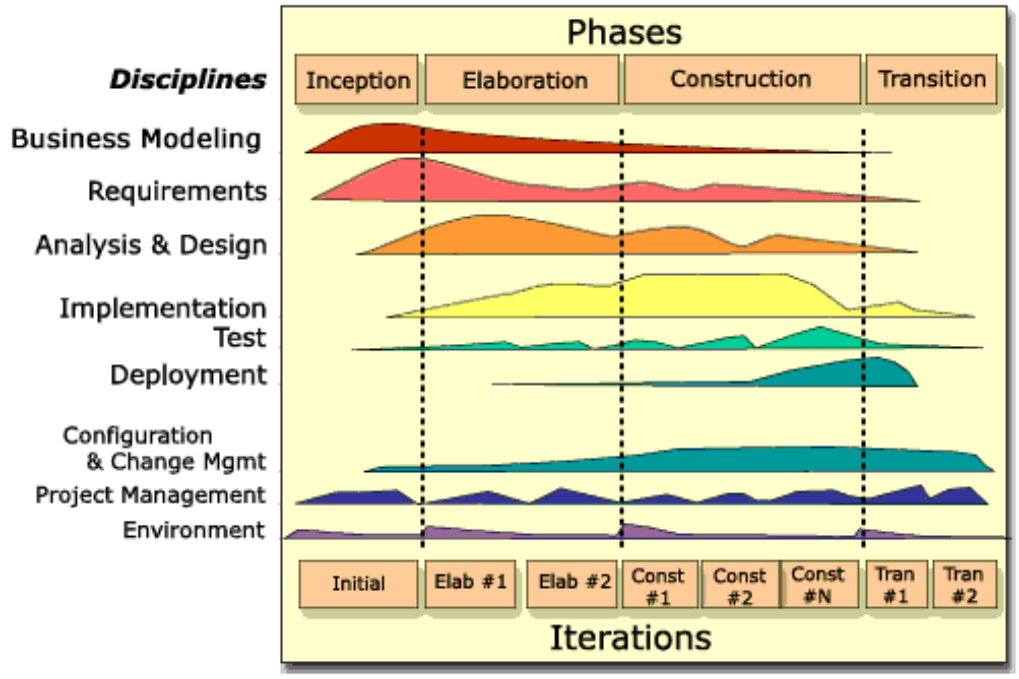

Figure 1: RUP disciplines and phases

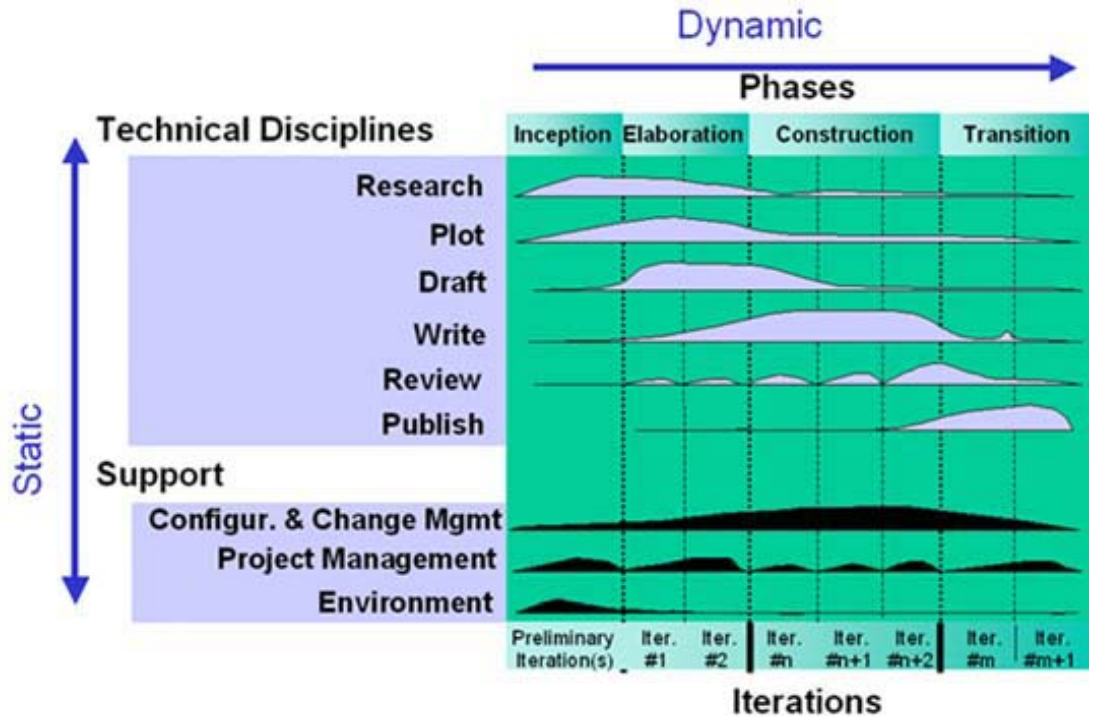

Figure 2: Mapping RUP to a writing project
Click to enlarge

Dynamic view: Phases

First, I assumed that the project was composed of four phases (as in RUP): Inception, Elaboration, Construction, and Transition. And, as in RUP, each phase had completion milestones, and some had artifacts as well.

Phase 1: Inception

During this phase I did the following:

1. Customized my own process.
2. Prepared my environment: the tools and infrastructure I needed for the project.
3. Defined the scope of the book and analyzed the target audience.
4. Created a Vision Document and obtained reviewer agreement on the subject matter.
5. Created a short description of all the chapters.
6. Created a more detailed description of critical chapters.

Artifacts: Vision Document, simple Project Plan (containing a Work Breakdown Structure [WBS] of the activities, list of risks, and role description of the reviewers involved in my project), and Requirements Management Plan.

Milestones: Develop a clear vision, set up tools, and prepare environment.

Phase 2: Elaboration

During this phase, I did the following:

1. Built the blueprint -- a synopsis that included a first version of the most critical parts of the book.
2. Collected and prepared information for all the other chapters.
3. Had the reviewers read the most important chapters and provide feedback.

Artifacts: Update Vision Document and Project Plan; create first version of critical chapters and storyboard of all other chapters.

Milestone: Obtain positive feedback from expert reviewers on the most important parts of the book.

Phase 3: Construction

In this phase I did the following:

1. Wrote all the text and sent it to editorial reviewers.
2. Rewrote and polished chapters as necessary.
3. Used the last part of the phase for final reviewing and polishing.

Milestone: A final manuscript that is ready to send to prospective publishers.

Phase 4: Transition

In this phase I did the following:

1. Sent the book to ten publishers (potential project stakeholders). Assessed the results and tried more publishers.
2. Signed a contract with the publisher that accepted the manuscript. (If you are a self-publisher, you must design and format the book with a page layout tool.)
3. Revised the book according to the publisher's requests.
4. Checked and approved a proof copy of the book.
5. Supplied a short biography and other information to help the publisher market my book.
6. Advertised the book through a Web site; sent prepublication copies to journalists and people in my personal network.

Milestone: The book is ready for distribution and sale to readers.

Static view: Disciplines

As you can see in Table 1 below and in Figure 2 above, I also built analogies between my nonsoftware process and a software process through RUP disciplines.

Table 1: Mapping RUP disciplines to the writing process

RUP technical disciplines	Writing discipline	Writing process
Business Modeling	Research	Research the topic of the book: investigate library materials, search the Web, and read various references. Use page markers. Read and classify notes and books.
Requirements	Plot	Define the vision of your project. Outline the structure or basic argument that you will develop. Have a clear idea of what to write.

Analysis and Design	Draft	Choose a writing style suitable for your audience and purpose. Define a format for the manuscript. Gather and filter the information you collected during the research stage. Identify main ideas and write drafts.
Implementation	Rewrite	Rewrite and polish the text. Revise in response to reviewers' suggestions. Correct grammar, spelling, and inconsistencies. Register data such as interesting quotations (and recipes in this case) within the IBM Rational RequisitePro* database creating a reusable repository. Generate texts with IBM Rational SoDA** templates, extracting iterative information from the Rational RequisitePro database.
Deployment	Publish	Prepare a campaign to find potential publishers and send manuscripts to them. Design the book with the publisher. Once you have a contract, manage the relationship with the publisher and meet your deadlines. Advertise and post portions of the book through a personal network and Web sites related to the book's subject. Send prepublication copies to journalists and key people in your personal network.
Test	Review	Take a break, and then go back and review your book again. Collect new ideas from reviewers and readers. Generate statistics about readability and quotes from reviewers to supply to the publisher's marketing department.
RUP support disciplines* **		**Writing process**
Project Management		Create an overall project plan and plan phases. Monitor progress against deadlines. Identify and mitigate risks (see section on Activity 4 below).

Configuration and Change Management		Create a data repository. Organize artifacts and notes to maintain integrity and attribute sources correctly. Make backup copies before beginning each revision.
Environment		Create a Development Case describing the process you will use. Decide how you will use automated and manual tools, and prepare the environment. Create MS Word templates to format your pages and IBM Rational SoDA templates to extract data from your tool databases.
***These **support disciplines** are universal to all projects.		

Best practices and tools view

In writing my book, I paid particular attention to four of the six RUP best practices:

- Develop iteratively.
- Manage requirements.
- Continuously verify quality.
- Manage changes.

Below, I will describe how these came into play in my project.

Develop iteratively

In planning, I divided my book into clearly identified chapters, which are in some ways similar to software project use cases. So, like the project manager and architect in RUP, I followed the lifecycle for use-cases: I outlined the chapters, prioritized all the topics I would cover in the chapters collectively, and then began to detail (develop) these topics. I ordered the topics by level of difficulty and importance. After I had done my research, I started with the topics at the top of the list -- those that were most difficult and important. I wrote first drafts, revised and polished them, and then proceeded to topics at the next level of difficulty. In this manner I developed the book iteratively. By tackling short and reachable objectives, I established a rhythm for my work and continued to feel creative rather than overwhelmed.

Just as iterations produce executable releases within a software project, so did my iterative process produce minor milestones: the first complete versions of chapters, ready to be polished.

Manage requirements

I decided not to jump immediately into writing. Instead I planned what to write before I started writing. However, as I wrote, new ideas and information emerged, so I always had to refer back to my original plans to see whether I was staying on track. My main organizational tool was IBM Rational RequisitePro, which enabled me to trace all my topics, drafts, and texts back to the artifacts I did during Inception and revise those as necessary.

Continuously verify quality

To help ensure high quality for my writing, I installed electronic dictionaries and used a tool named Cordial to analyze texts, correct grammar mistakes, and minimize awkward constructions. As I've already indicated, I also frequently reviewed and polished the texts in response to external reviewers' suggestions, and asked these reviewers to comment at the end of the Elaboration and Construction phases.

Manage changes

An automated configuration management tool can help writers manage the text as it evolves and keep track of each revision so that no data is lost. It can also help them evolve their planning documents. I used IBM Rational ClearCase to maintain my text and planning artifacts. Rational ClearCase can compare a current version of a Microsoft Word document with any previous version. This helped a great deal because my reviewers used Word's tracking functions when they commented on and edited my chapters.

Activity 2: Produce a development case

As I have said, I used artifacts to drive my writing and publishing process. My first step was to create a Development Case that outlined the main tasks required to produce these artifacts -- and eventually the book. Table 2 shows the Development Case, including phases, disciplines and artifacts, and tools I used to create the artifacts.

Table 2: Development Case: Artifacts, phases, disciplines, and tools

Phase				Discipline and its artifacts	Tools
I	E	C	T	Environment	
X				Development Case	Microsoft Word
X	X	X		Templates: Microsoft Word and SODA templates for IBM Rational RequisitePro, and separate templates to use within Word	Microsoft Word, IBM Rational RequisitePro, IBM Rational SoDA

I	E	C	T	Research	
X	X	X		Quotations and notes from reference books	IBM Rational RequisitePro
I	**E**	**C**	**T**	**Plan**	
X	X			Vision Document as a synopsis of the book	IBM Rational RequisitePro
X	X			Requirements Management Plan	IBM Rational RequisitePro
I	**E**	**C**	**T**	**Draft**	
X	X	X		Introduction	IBM Rational RequisitePro
X	X	X		Brief History of Moroccan Cooking	IBM Rational RequisitePro
X	X	X		Ten Major Ingredients for Moroccan Cooking	IBM Rational RequisitePro
X	X	X		How to Cook Moroccan Food	IBM Rational RequisitePro
X	X	X		Moroccan Cooking the World Over	IBM Rational RequisitePro
I	**E**	**C**	**T**	**Write**	
	X	X	X	Chapter 1: Introduction	Microsoft Word
	X	X	X	Chapter 2: Brief History of Moroccan Cooking	Microsoft Word
	X	X	X	Chapter 3: Ten Major Ingredients for Moroccan Cooking	Microsoft Word
		X	X	Chapter 4: From Ingredients to Recipes	IBM Rational SoDA, Microsoft Word
		X	X	Chapter 5: 100 Moroccan Recipes	IBM Rational SoDA, Microsoft Word
	X	X	X	Chapter 6: How to Cook Moroccan Food	Microsoft Word
	X	X	X	Chapter 7: Moroccan Cooking the World Over	Microsoft Word
	X	X	X	Chapter 8: References	Microsoft Word
	X	X	X	Chapter 9: Glossary	Microsoft Word
I	**E**	**C**	**T**	**Review**	
	X	X	X	Readability metrics and spelling	Cordial

	X	X		Reviewer approval	Microsoft Word
I	**E**	**C**	**T**	**Publish**	
		X	X	Final version to publisher	Paper
			X	Version revised and sent to publisher with suggestions and deadlines	Paper
			X	Final Layout from publisher ready to sent to printing	QuarkXpress
I	**E**	**C**	**T**	**Configuration and Change Management**	
	X	X	X	Remarks from reviewers	Microsoft Word (with Track Changes)
X	X	X	X	Project repository	IBM Rational ClearCase
I	**E**	**C**	**T**	**Project Management**	
X	X	X	X	Project Plan to reference and revise throughout the project	Microsoft Word

Activity 3: Develop a Requirements Management Plan

A Requirements Management Plan describes requirement types and links between requirements. It summarizes the specific organization, based on input from IBM Rational RequisitePro, which stores requirements by type, either within a database or within both Microsoft Word documents and a database. Table 3 shows the requirement types for my book.

Table 3: Requirement types for book

Artifact (Document Type)*	Requirement Type (Prefix)**	Description
Vision (VIS)	Main objectives & contents (FEAT)	Includes synopsis, main objectives and ideas to share throughout the book, contents, and market opportunities for the publisher.
Storyboard (SRB)	Drafts (SR)	Drafts of chapters, incorporating research.
	Ingredients (ING)	Information about the major ingredients used in the recipes.

	Text to be sent to the editor (TXT)	The text document is inserted into the draft document with a hyperlink. See Figure 3 below.
Glossary (GLS)	Terms (TERM)	Glossary of specific terms used within the book.
N/A	Recipes (RECP)	Recipes are managed in the IBM Rational RequisitePro database and will be printed in specific formats or tables. IBM Rational SoDA will be used to generate text within Microsoft Word.
Quotation (QUO)	Quotations (QUOT)	Quotations and important notes from external sources are registered in the IBM Rational RequisitePro database.
Reference (REF)	Books and references (REFR)	Source information about articles and books that I used are registered in the IBM Rational RequisitePro database.
N/A	Photos and images (PHO)	Photos and images are managed in the IBM Rational RequisitePro database. They can be accessed through a Rational RequisitePro attribute (defined as a URL) before being inserted into the text.

*A ***document type*** is a template that is applied to IBM Rational RequisitePro documents. For instance, the Vision Document is named Vision.vis within Windows.

A ***requirement type* is a template for requirements. Requirement types are used to classify similar requirements; the prefix is useful to distinguish each type of Rational RequisitePro requirement when displaying views. For requirement types, you define a common set of attributes, display style, and tag numbering. Attributes such as "status" are helpful to sort requirements with queries.

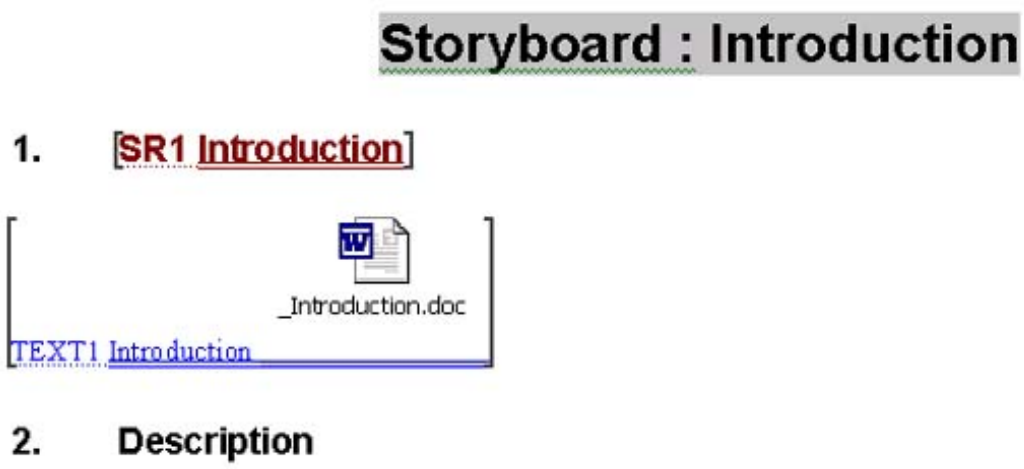

Figure 3: Example implementation of a text (TXT) requirement within a storyboard (SRB) document

Traceability

IBM Rational RequisitePro provides the ability to trace a project requirement element to other related project requirement elements. For my book, I used this capability in the following ways:

- Tracing content plans to drafts.
- Accessing texts within drafts via links inserted with Microsoft Word.
- Tracing all major ingredients to recipes.
- Tracing quotations within texts to references.
- Including photos and images in texts.

Figure 4 shows my traceability model with its main attributes.

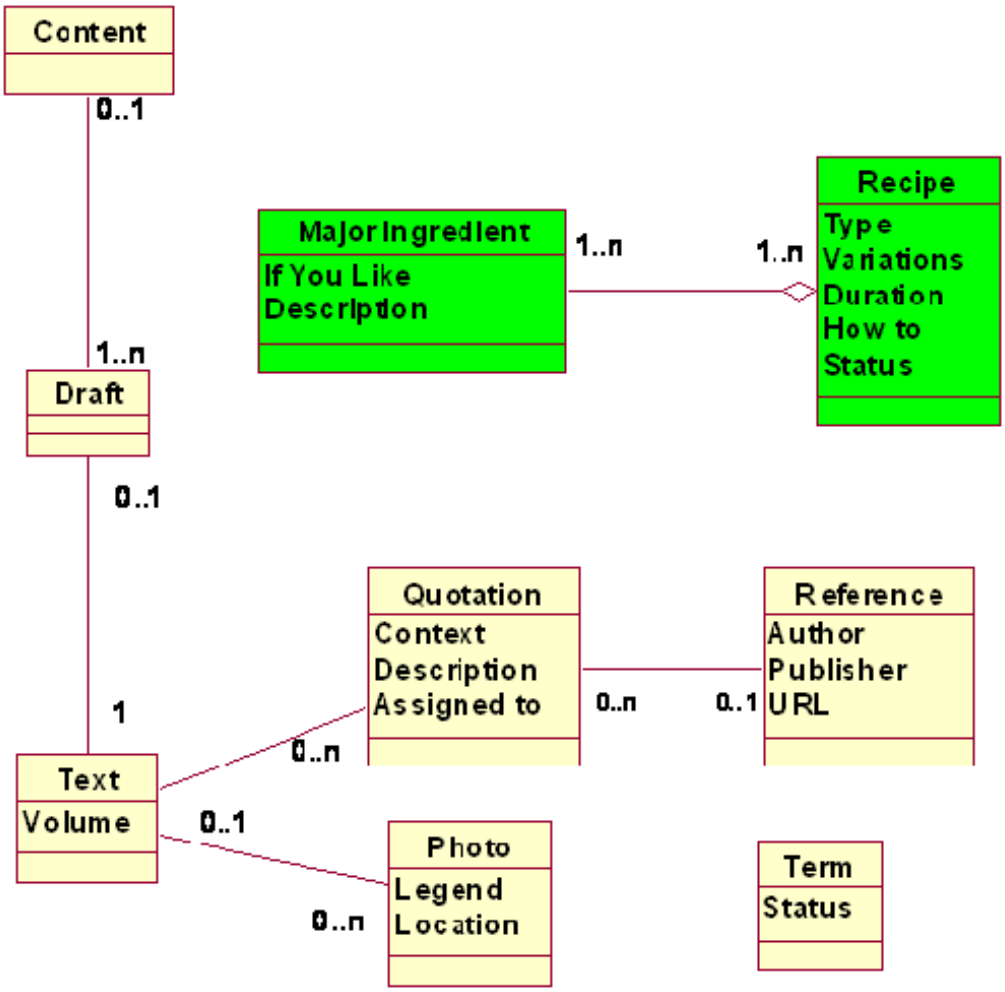

Figure 4: Traceability model for handbook of Moroccan cooking, with main attributes

Activity 4: Identify and Mitigate Risks

A key principle of RUP is to attack major risks as soon as you discover them. Below I describe the three major risks I identified and explain the mitigation strategies I developed for them.

Risk #1: Factual errors. Although I will try not to, I may make erroneous statements and/or include irrelevant material.

Mitigation strategy: Doing disciplined research, carefully noting my sources, and cross-checking facts will help me ensure that my statements and assumptions are correct. The Elaboration phase ends when expert reviewers have read the critical chapters and have provided positive feedback. During the Construction phase, I may undertake practical work (such as testing recipes) to validate some hypotheses.

Risk #2: No publisher. There is a chance that the book will not be accepted by a publisher.

Mitigation strategy: As soon as possible, I will send query letters to potential publishers to ensure that I am on the right track with my vision for the book. If no one shows any interest, or if I don't find a publisher after I finish the manuscript, I will publish the book by myself (i.e., become a self-publisher), with sponsorship from an association. It will be worth the investment: The association will advertise the book on its Web site, which will ensure some sales.

Risk #3: Project overload and delay. The project might get out of control if there is endless work and rework; I might grow discouraged and fall behind on the schedule I set for myself or that the publisher sets for me.

Mitigation strategy: During the Elaboration phase, I will write the most complex and critical parts of the book. Early reviewer feedback and revisions will give me confidence that the book is worthwhile, and that I can complete it. Tracking progress through minor and major milestones will help me to stay focused and motivated.

Using IBM Rational RequisitePro

I've already discussed many specific ways I applied IBM Rational RequisitePro in this project, but it is worth looking at three basic ways writers can leverage this product:

- By using database power to immediately access pertinent information.
- By connecting data through the power of traceability.
- For measuring progress from day to day.

Using database power

Every day, I start my writing work by clicking on IBM Rational RequisitePro for a clear overview of the project structure (see Figure 5). With two or three mouse clicks, I can open the Microsoft Word document of my choice and begin developing it with information and ideas I've captured in the database.

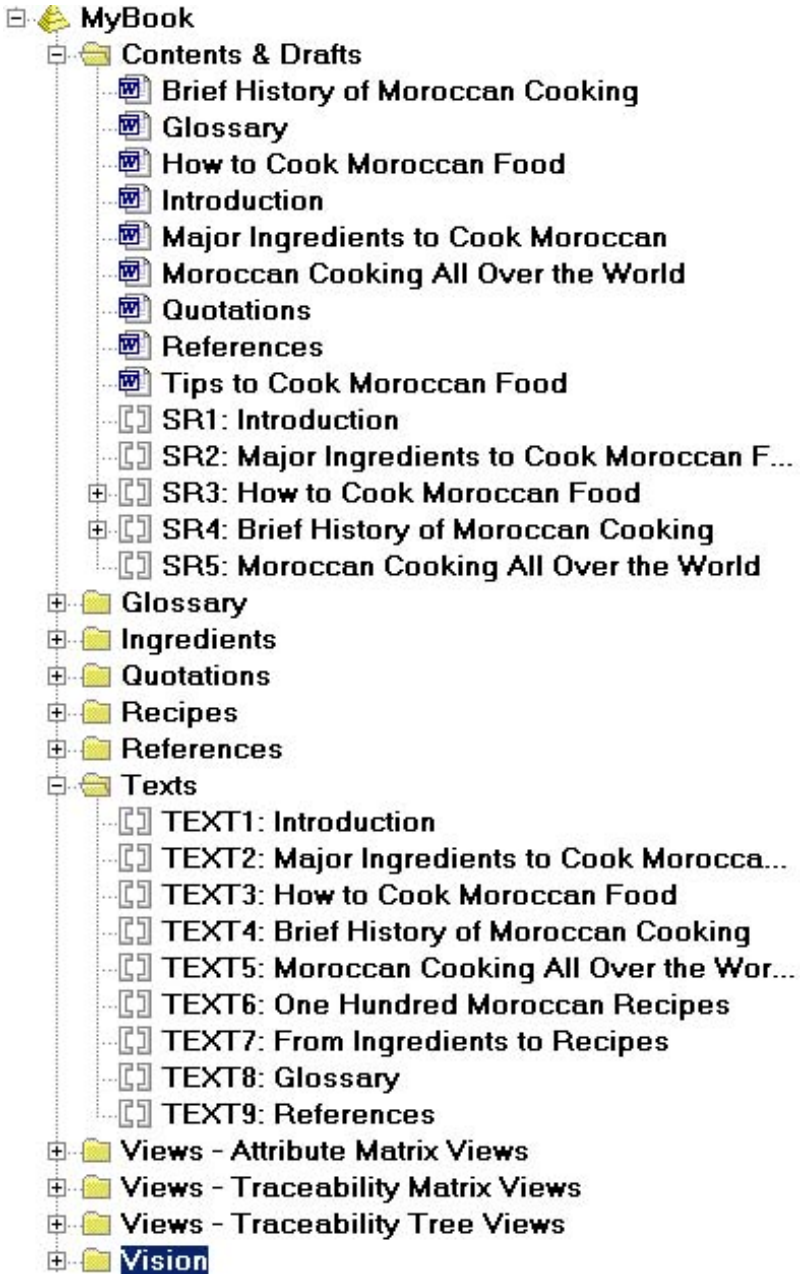

Figure 5: Viewing the project structure in IBM Rational RequisitePro

I have the ability to manage and easily locate these documents through views based on database information such as attributes and traceability relationships (see Figure 6). And I can register information I synthesize and summarize information within the Rational RequisitePro database as well. If I need to query or sort any data, I can go directly to the critical information stored in this database instead of searching at random through directories and documents.

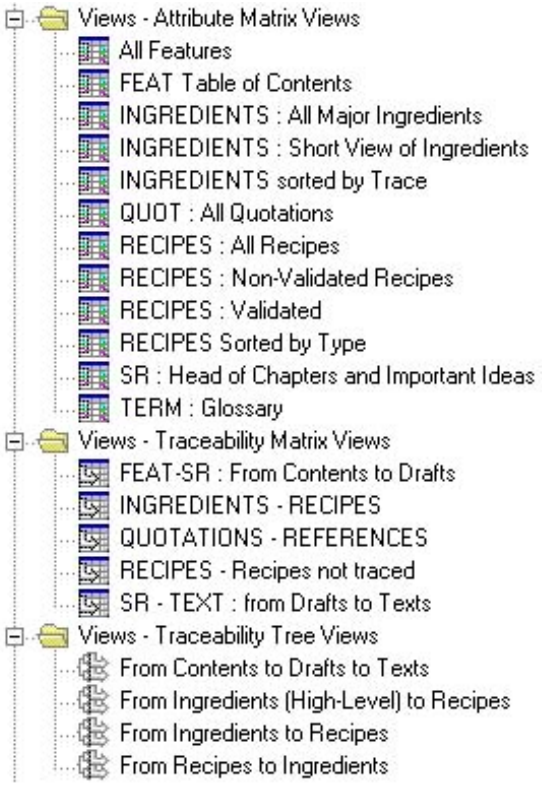

Figure 6: Locating information through IBM Rational RequisitePro Views

Connecting data with traceability

You might think that traceability is less useful for a personal writing project than it is for a large team software development effort, but that is not the case. Explicit traceability helps authors manage complex information; for instance, I may want to select one hundred recipes with ten major ingredients, or vice versa. As I create these recipes, the database will help me determine, via the status attribute, whether each recipe is validated. With traceability, I can connect ingredients to recipes and show recipes that are not yet traced from Ingredients. And finally, using the built-in capabilities of IBM Rational SoDA, I can generate a list of recipes sorted by ingredients. Figure 7 shows the traceability matrix I used to generate this list.

Relationships: - direct only	RECP2: Lamb Tagine with Artichokes, Lemon and...	RECP3: Meatball, Tomato and Egg Tagine	RECP4: Large-Grain Couscous (Mhammas)
⊟ INGR1: Lemon!			
INGR1.1: Preserved...			
INGR1.2: Lemons	↱		
⊟ INGR2: Lamb !			
INGR2.1: Meatball		↱	
INGR2.2: Kefta			
INGR2.3: Braised	↱		
INGR2.4: Stuffed			
⊟ INGR3: Chicken			
INGR3.1: Stuffed		↱	
INGR3.2: Braised			
INGR3.3: Chicken			
⊟ INGR4: Couscous !			
INGR4.1: Fine-Grain			
INGR4.2: Large-Grain!			↱
⊟ INGR5: Fish			
INGR5.1: Stuffed			
⊟ INGR6: Olives !			
INGR6.1: Green			
INGR6.2: Black			
⊟ INGR7: Egg !			
INGR7.1: hard-boiled !			
INGR7.2: soft!		↱	
⊟ INGR8: Fruits!			
INGR8.1: Prunes			
INGR8.2: Dates !			
INGR8.3: Raisins	↱		

Figure 7: Traceability matrix: Ingredients to recipes

Measuring progress from day to day

As I worked on my project, it was useful to know how many recipes had been validated and how many recipes there were of each type (e.g., salads, meat, sweets, etc.). IBM Rational RequisitePro generates simple metrics (such as those shown in Figure 8) to assess progress along such lines.

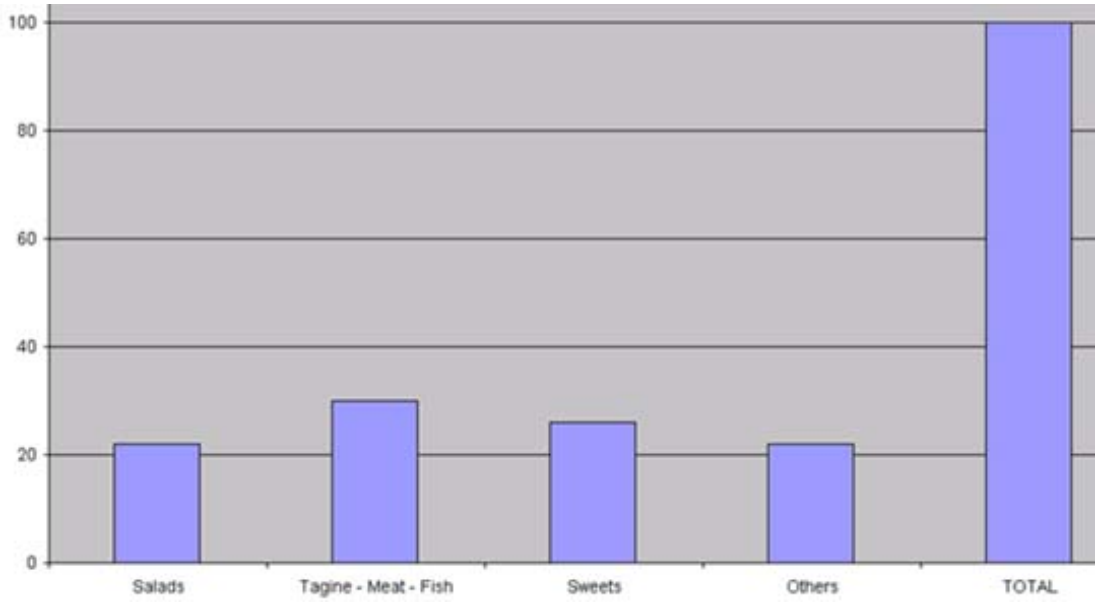

Figure 8: Count of validated recipes by type
Click to enlarge

Evaluating results

Up to the Transition phase, my cookbook project had taken approximately eight weeks of effort, distributed over an eighteen-month period. Early in the Inception phase, while creating my work environment, I asked myself: "Why don't you just write your book instead of taking all this time to set up the environment?" But thanks to the tools I was using, three weeks later, at the end of the Elaboration phase, I had a much clearer view of what was left to do than I would have had if I had not done that planning work up front. By using tools, I had built an efficient "machine" to successfully produce and finalize the book. During the Construction phase this machine helped me to maintain integrity and consistency in my texts and to stay focused on objectives I identified at the beginning of the project.

When the book is published, what will I do with the Rational RequisitePro database? I'll continue to extract information through different points of view to write articles on specific topics related to Moroccan cooking. And if I keep adding to the database, I'll be ready to publish the second volume of my book!

Tools made a difference

Without RUP, as a beginning writer, I would have tried to write my book in sequential mode. Instead, using RUP, I outlined my vision, gathered essential information, and prioritized my topics before starting to write. I also thought about risks and difficulties. I wrote the critical parts early on, during the Elaboration phase and had them reviewed by experts, and that meant I could do my polishing during the Construction phase. Plus, my early success with the writing gave me confidence and reinforced my willingness to finish, despite the huge task ahead. Applying the same tools I use every day as a software engineer helped me take a professional

approach to my work in a completely different domain. Thanks to RUP, IBM Rational RequisitePro, and IBM Rational SoDA, I completed my first book with the level of quality I expected.

Acknowledgments

Special thanks to Catherine Southwood and Marlene Ellin for their patience and encouragement in helping me transform my draft into a readable paper.

Notes

[1] IBM Rational RequisitePro is a flexible tool used for documenting, organizing, and managing requirements. A writer can use it to store important information, ideas, and chapter titles in a central repository; to provide quick access to all project artifacts; and to query and sort information throughout the project lifecycle.

IBM Rational SoDA automates the creation and maintenance of project documentation and reports. It generates documents by extracting data from multiple sources and translating it into Microsoft Word. I used it to create text from information stored within my IBM Rational RequisitePro database.

IBM Rational ClearCase is a software configuration management (SCM) tool that provides advanced version control support, with dynamic views of distributed assets and versions.

For more information on the products or services discussed in this article, please click here and follow the instructions provided. Thank you!