Presentation & Requirements

Database and Web Applications Laboratory

Sérgio Nunes/João Correia Lopes

INESC TEC – Faculdade de Engenharia, Universidade do Porto

15 February 2019

GOALS

By the end of this class, the student should be able to:

- Describe the content, evaluation and bibliography of the course
- Use tools for collaborative development of documentation
- Gather the specification of the actors and user stories of a system
- Gather the specification of supplementary requirements of a system

TIPS

- The slides display a brief summary of the theory that does not replaces **reading the bibliography** listed in the class sheet
- “Students are responsible for anything that transpires during a class—therefore **if you’re not in a class**, you should get notes from someone else (not the instructor)”—David Mayer
- The best thing to do is to **read carefully** and **understand** the documentation published in the course web site (or else **ask** in the class)
- We will be using **Moodle** as the primary means of communication

Part I

Presentation of the Course

Contents

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/`

FACT SHEET

- Skills and learning outcomes
- Program
- Bibliography
- Evaluation
- Final grade

$$Final\ grade = 80\%\ NP + 20\%\ NI$$

$$NP = 20\%\ ER + 20\%\ EBD + 20\%\ EAP + 40\%\ PA + \Delta_{ind}$$
$$\Delta_{ind} \in [-10\%, 10\%]$$

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/sheet`

PLAN OF THE COURSE

- Lectures
- Practical classes
- Assessment dates

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/plan`

PRACTICAL WORK BY-LAWS

- Practical work
- Documentation
- Technologies
- Assessment

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/bylaws`

WRITING DOCUMENTATION

- See the published requirements
- Follow the template given
- Generate a PDF and upload to the Moodle of the group

⇒ Templates

WEB ARCHITECTURE

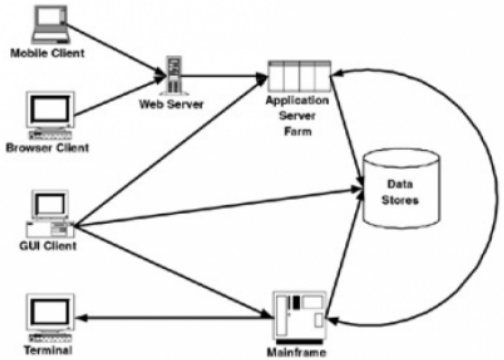


Figure 14.17: Potential deployment architecture for modern applications.

Scott Ambler. The Object Primer. Cambridge University Press, 3rd Edition, 2004 (chap. 14)

⇒ stack-overflow-the-architecture-2016-edition

⇒ February 2015 upgrade entire 256 image album

SOFTWARE DEVELOPMENT

- Planning
- Analysis
- Design
- Development
- Testing
- Deployment
- Maintenance and Operation

DESIGN OF A DATABASE

- Conceptual Design
- Logical Design
- Physical Design

LIST OF ARTEFACTS

20 ER: Requirements specification and User interfaces
- 2 A1: Project presentation
- 6 A2: Actors and User stories
- 12 A3: User Interfaces Prototype

20 EBD: Database specification
- 10 A4: Conceptual Data Model
- 4 A5: Relational Schema, validation and schema refinement
- 6 A6: Integrity constraints, indexes, triggers and user functions, database populated with data

20 EAP: Architecture specification and Prototype
- 9 A7: High-level architecture. Privileges. Web resources specification
- 8 A8: Vertical prototype
- 3 A9: Main accesses to the database and transactions

40 PA: Product and presentation
- 35 A10: Product
- 5 A11: Presentation and discussion

LIST OF ARTEFACTS

20 ER: Requirements specification and User interfaces
- 2 A1: Project presentation
- 6 A2: Actors and User stories
- 12 A3: User Interfaces Prototype

20 EBD: Database specification
- 10 A4: Conceptual Data Model
- 4 A5: Relational Schema, validation and schema refinement
- 6 A6: Integrity constraints, indexes, triggers and user functions, database populated with data

20 EAP: Architecture specification and Prototype
- 9 A7: High-level architecture. Privileges. Web resources specification
- 8 A8: Vertical prototype
- 3 A9: Main accesses to the database and transactions

40 PA: Product and presentation
- 35 A10: Product
- 5 A11: Presentation and discussion

LIST OF ARTEFACTS

20 ER: Requirements specification and User interfaces
- 2 A1: Project presentation
- 6 A2: Actors and User stories
- 12 A3: User Interfaces Prototype

20 EBD: Database specification
- 10 A4: Conceptual Data Model
- 4 A5: Relational Schema, validation and schema refinement
- 6 A6: Integrity constraints, indexes, triggers and user functions, database populated with data

20 EAP: Architecture specification and Prototype
- 9 A7: High-level architecture. Privileges. Web resources specification
- 8 A8: Vertical prototype
- 3 A9: Main accesses to the database and transactions

40 PA: Product and presentation
- 35 A10: Product
- 5 A11: Presentation and discussion

LIST OF ARTEFACTS

- 20 ER: Requirements specification and User interfaces
  - 2 A1: Project presentation
  - 6 A2: Actors and User stories
  - 12 A3: User Interfaces Prototype
- 20 EBD: Database specification
  - 10 A4: Conceptual Data Model
  - 4 A5: Relational Schema, validation and schema refinement
  - 6 A6: Integrity constraints, indexes, triggers and user functions, database populated with data
- 20 EAP: Architecture specification and Prototype
  - 9 A7: High-level architecture. Privileges. Web resources specification
  - 8 A8: Vertical prototype
  - 3 A9: Main accesses to the database and transactions
- 40 PA: Product and presentation
  - 35 A10: Product
  - 5 A11: Presentation and discussion

WHAT WE LEARNED

- The content, evaluation and bibliography of the course
- The content and form of the project documentation

Part II

Requirements Specification

CONTENTS

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/artefacts_er`

PROJECT PRESENTATION (A1)

A1

This artefact introduces the context and motivation and briefly describes the Web information system to be developed.
It includes the goals of the project and lists the features that should be supported, together with the identified access groups.

⇒ `https://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/medialib/a1`

USAGE MODELLING

You need to understand how people will work with your system.

- An important part of software development is to explore the requirements for your system.
- Usage modelling explores how people work with a system, vital information that you require if you are going to successfully build something that meets their actual needs.
- You cannot successfully build a system if you do not know what it should do, and a critical aspect of this is exploring how people will actually use the system.

Scott Ambler. The Object Primer. Cambridge University Press, 3rd Edition, 2004 (chap. 5)

Usage Modelling

You need to understand how people will work with your system.

- An important part of software development is to explore the requirements for your system.
- Usage modelling explores how people work with a system, vital information that you require if you are going to successfully build something that meets their actual needs.
- You cannot successfully build a system if you do not know what it should do, and a critical aspect of this is exploring how people will actually use the system.

Scott Ambler. The Object Primer. Cambridge University Press, 3rd Edition, 2004 (chap. 5)

ACTORS AND USER STORIES (A2)

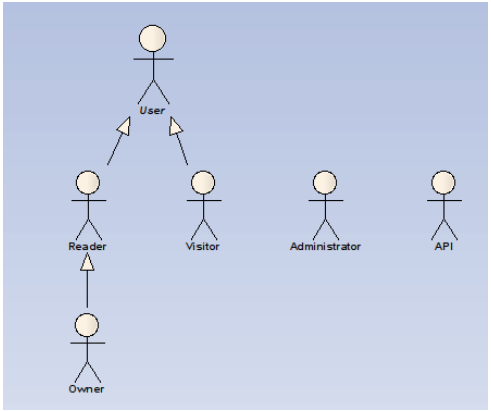
A2

This artefact contains the specification of the actors and their user stories, serving as agile documentation of project requirements.

⇒ `http://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/artefacts_er`

ACTORS

- An actor is any entity that interacts with the system being specified
- Persons, and other organizations external to the system
- Actors are always external to the system being modelled



DESCRIPTION OF ACTORS

Id.	Description	Examples
User	Generic user that has access to public information, such as collection's items	n/a
Visitor	Unauthenticated user that can register itself (sign-up) or sign-in in the system	n/a
Reader	Authenticated user that can consult information, insert works and items, manage list of interests, request the loan of items and comment the works of the collection	jlopes
Owner	Authenticated user that belongs to the same location as the creator of an item and can change the existing information or lend and record the return of items	jlopes
Administrator	Authenticated user that is responsible for the management of users and for some specific supervisory and moderation functions	admin
OAuth API	External OAuth API that can be used to register or authenticate into the system	Google

⇒ `https://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/medialib/a2`

USER STORIES

- US are a high-level definition of a requirement
- US are a brief description of a potential interaction with the system by one of its users
- US focuses only on interaction requirements and not on the technical aspects of the design of the system
- US have a priority (high or essential, medium or conditional, and low or optional) indicating the need to be included in the design implementation stages and project
- A team of expert analysts also includes an estimate of the effort required for its implementation

Scott Ambler. The Object Primer. Cambridge University Press, 3rd Edition, 2004 (chap. 5)

USER STORY TEMPLATE

TEMPLATE

"As a [user], I want [function], so that [value]"

Steve Dennis, How to write meaningful User Stories, subcite.com Articles & Tutorials, 2010

LIST OF USER STORIES

Id.	Name	Priority	Description
US01	Sign-in	high	As an Visitor, I want to authenticate into the system, so that I can access privileged information
US02	Sign-up	high	As Visitor, I want to register myself into the system, so that I can authenticate myself into the system
US03	Sign-up using external API	low	As a Visitor, I want to register a new account linked to my Google account, so that I can access privileged information
US04	Sign-in using external API	low	As a Visitor, I want to sign-in through my Google account, so that I can authenticate myself into the system
US11	Home page	high	As an User, I want to access home page, so that I can see a brief website's presentation
US12	About page	high	As an User, I want to access the about page, so that I can see a complete website's description
US16	Search	high	As an User, I want to consult all the public information, so that I be informed
US17	Statistics	medium	As an User I want to check usage statistics (for example the top of most requested works), so that I stay informed

⇒ `https://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/medialib/a2`

SUPPLEMENTARY REQUIREMENTS

In addition to functional requirements there are other restrictions to the project: business rules, technical requirements

- A business rule defines or constrains one aspect of the business, with the intention of asserting business structure or influencing business behaviour
- Technical requirements are concerned with the technical aspects that the system must meet, such as performance-related issues, reliability issues and availability issues
- A restriction on the design limits the degree of freedom in the search for a solution

Scott Ambler. The Object Primer. Cambridge University Press, 3rd Edition, 2004 (chap. 7)

BUSINESS RULES

Id.	Name	Description
BR01	Ownership	Only a user from the same site that the creator of the item (Owner) can lend or register your return
BR02	Return Date	The return date must be greater than the loan date for one item that has not yet been returned

⇒ https://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/medialib/a2#business_rules

TECHNICAL REQUIREMENTS

Id.	Name	Description
TR02	Accessibility	The system must ensure that everyone can access the pages, regardless of whether they have any handicap or not, or the Web browser they use
TR03	Usability	The system should be simple and easy to use
TR04	Performance	The system should have response times shorter than 2s to ensure the user's attention
TR05	Architecture	The system should be implemented as a Web application with dynamic pages (HTML5, JavaScript, CSS3 and PHP)
TR07	Database	The PostgreSQL 9.4 database management system must be used
TR08	Security	The system shall protect information from unauthorised access through the use of an authentication and verification system
TR11	Ethics	The system must respect the ethical principles in software development (for example, the password must be stored encrypted to ensure that only the owner knows it)

⇒ `https://web.fe.up.pt/~jlopes/doku.php/teach/lbaw/medialib/a4`

WHAT WE LEARNED

- Project presentation (A1)
- Actors and user stories of a system (A2)