



**FACULTY OF ARCHITECTURE AND ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**



**TestMe Requirements Specification**

**Version 1.0**

**April 19, 2021**

**Group: IIIB**

**Team members: Martina Kajana, Kleit Daja, Raziena Uruci, Imran Sulo, Redis Kondakciu, Joana Allushi.**

**Project name: TestMe**

# Table of Contents

<b>1. EXECUTIVE SUMMARY</b>	<b>3</b>
1.1 PROJECT OVERVIEW	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION	3
<b>2. PRODUCT/SERVICE DESCRIPTION</b>	<b>3</b>
2.1 PRODUCT CONTEXT	3
2.2 USER CHARACTERISTICS	3
2.3 ASSUMPTIONS	3
2.4 CONSTRAINTS	3
2.5 DEPENDENCIES	4
<b>3. REQUIREMENTS</b>	<b>4</b>
3.1 FUNCTIONAL REQUIREMENTS	5
3.2 NON-FUNCTIONAL REQUIREMENTS	5
3.2.1 <b>Product requirements</b>	5
3.2.1.1 Usability	
3.2.1.2 Efficiency	
3.2.1.3 Dependability	6
3.2.1.4 Security	6
3.2.2 <b>Organizational Requirements</b>	7
3.2.2.1 Environmental Requirements	8
3.2.2.2 Operational Requirements	8
3.2.2.3 Development Requirements	8
3.2.3 <b>External Requirements</b>	8
3.2.3.1 Regulatory Requirements	
3.2.3.2 Ethical Requirements	
3.2.4 <b>Legislative Requirements</b>	
3.2.4.1 Accounting Requirements	
3.2.4.2 Safety Requirements	
3.3 DOMAIN REQUIREMENTS	9
<b>4. USER SCENARIOS/USE CASES</b>	<b>9</b>
<b>APPENDIX</b>	<b>10</b>
APPENDIX A. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	10
APPENDIX B. REFERENCES	10
APPENDIX C. REQUIREMENTS TRACEABILITY MATRIX	10
APPENDIX D. ORGANIZING THE REQUIREMENTS	12

# 1. Executive Summary

## 1.1 Project Overview

TestMe is a new project that concerns online examination. The project will consist of a java application and a webpage that advertises it. The program can be used by anyone who is interested in boosting their academic career and skills.

The application will include exams of different subjects during a specified time frame. This way anyone can get a certification of any specific topic without the need of any previous university diploma. With just some simple configuration, teachers can create an exam using multiple choice questions or open questions.

### ***Purpose and Scope of this Specification***

Nowadays, everything and everyone is constantly online. People find it easier and more comfortable to sort things out through technology. Because of this and because of how covid-19 has impacted our everyday life, we thought of the idea of a program that enables the online examination.

The purpose of our project is to test the subject knowledge of the students. It will conduct digital exams and will make the evaluation of the students' academic knowledge easier. With some simple configuration, teachers & professors can create an exam using multiple choice questions or open questions. This method eliminates the drawbacks in the traditional mode of the pen-and-paper examination. People can conduct exams anytime, anywhere, while preserving the exam's credibility and integrity.

This project has many benefits as further mentioned:

### **THE BENEFITS:**

1. Automatic evaluation – the process of evaluation is easier and faster because it is automated. Students like it better because they do not have to wait too long for the results. Teachers prefer it too, because it is less time consuming than the traditional way.
2. Cost effective – When taking conventional exams, there is a higher consumption of question papers and answer sheets than online examination. Thus, online exams reduce these expenses.
3. Respects social distancing (covid-19) – we are living through a pandemic and everything is unpredictable. Online exams prevent the spread of the virus and protect our lives.
4. It is more secure – since the exam is stored in a database there is no chance for a paper leakage. Also the system will store the exam directly and it can be visible only to the student who took it in anytime.
5. Motivation – each time a student correctly completes more than 85% of the exam, they are awarded with a certificate. We want this certificate to be a source of motivation for every learner. Studies show that when certificates are removed from the learning platform, exam results drop by almost 50%.

## 2. Product/Service Description

### 2.1 **Product Context**

The project will offer an online examination program where students can take exams and a web page which will offer information about the program. The program will have three types of users: teacher, student and administrator. The teacher will be able to declare the right answers of a test so that the evaluation can be automatically done. The students will be able to take exams and the administrator will be able to handle any inconvenience. The website will contain information about the program, the exams, exam's date, the students' certificates, and information about teachers. People can log in and get updates on the exams.

This online examination program will offer students from all over the world, the chance to take exams just by sitting in front of the computer. It aims for the students to increase their knowledge by learning from their mistakes. The certificate is also added as a way to motivate the student. We expect that the student will take the examination more seriously if we put a 'price' such as a certificate of achievement.

### 2.2 **User Characteristics**

The users of the application will be: Teacher, Student and Admin.

#### **TEACHER :**

- Teachers can log in to the application as a 'teacher'.
- Teachers can enter the teacher management section.
- Teachers can enter the questions in the question banks.
- Teachers have to determine the correct answers of the test.
- Teachers can have access to the students' scores.
- Teachers can evaluate open questions.
- Teachers can see the students' examination time in the teacher management page.

#### **STUDENT :**

- Students can log in to the application as a 'student'.
- Students have to decide in which profile they are interested to take the exam (social exam, science exam).
- Students can start the exam when they click the start button.
- Students can take the exam within the given time.
- After the end of the exam the students will know if they earn a certificate or not.
- The students can see their scores and where they answered the questions wrong.

#### **ADMIN :**

- Admins can log in to the application as an 'admin'.
- Admins have access to the data of registered teachers and students.
- Admins have access to students' results.
- Admins are responsible for the certificates granted in each subject.

### 2.3 Assumptions

List any assumptions that affect the requirements, for example, equipment availability, user expertise, etc. For example, a specific operating system is assumed to be available; if the operating system is not available, the Requirements Specification would then have to change accordingly.

### 2.4 Constraints

Describe any items that will constrain the design options, including

- parallel operation with an old system
- audit functions (audit trail, log files, etc.)
- access, management and security
- criticality of the application
- system resource constraints (e.g., limits on disk space or other hardware limitations)
- other design constraints (e.g., design or other standards, such as programming language or framework)

### 2.5 Dependencies

List dependencies that affect the requirements. Examples:

- This new product will require a daily download of data from X,
- Module X needs to be completed before this module can be built.

## 3. Requirements

### Functional Requirements

In the example below, the requirement numbering has a scheme - BR\_LR\_0## (BR for Business Requirement, LR for Labor Relations). For small projects simply BR-## would suffice. Keep in mind that if no prefix is used, the traceability matrix may be difficult to create (e.g., no differentiation between '02' as a business requirement vs. a test case)

The following table is an example format for requirements. Choose whatever format works best for your project.

For Example:

Req#	Requirement	Comments			
BR_01	The online examination program, once opened, should have two options: 1. Log in 2. Sign up	The two options are available for whoever is registered in the 'TestMe' program (log in) and for who is not (sing up).			
SIGN UP					

### TestMe Requirements Specification

BR_02	The sign up option will allow the first time user to create an account as a student or as a teacher.	The sign up can be completed with the completion of a form: name, surname, email, password, phone number and specify whether they are a teacher or a student. Once signed up, the teacher/ student can always log in through this account.			
<b>LOG IN</b>					
BR_03	The online examination program should have <b>teacher</b> login.	The log in teacher will include: teacher username and teacher password. After entering the username and password, the teacher logs in the teacher management section.			
BR_04	The online examination program should have <b>student</b> login.	The log in student will include: student username and password. After entering the username and password, the student logs in to the examination page.			
BR_05	The online examination program should have <b>admin</b> login.	The log in admin will include: admin username and password. The admin will enter the BTS page (Behind The Scene). He will have access to: teacher's data and student's data.			
<b>Behind The Scene BTS section</b>					
BR_06	The online examination program should have a Behind The Scene section.	BTS includes: data of teachers and students, student scores and examination time.	2		

### TestMe Requirements Specification

BR_07	The Behind the Scene page should include: Certificate of Achievement page (CoA).	CoA should include certificates for each kind of exam subject the program will have. If the student meets the condition for a certificate, a certificate will automatically be generated after the results have been shown.	2 3		
BR_08	Once the student starts the exam, the examination process begins.	After choosing the desired subject, the student is asked for the confirmation of the subject. Then, the student starts the exam and the timer is set. If the student does not complete the exam during the required time, the session times out.			
BR_09	After the examination process ends, the result will appear on the screen.	When the student wants to finish the exam, he/she is asked for confirmation. After the confirmation, the exam will be closed and on the screen the student will see his/her score.			
BR_10	After the results are shown, the students will receive the certificate or not.	Those who have completed 85% of the exam correctly will receive the certificate of achievement.			
<b>Review Section</b>					
BR_11	The online examination program should include a 'Review' section.	After the exam is finished, the students may have access to the review section. This section will contain: the questions and answers of the student's exam, and his/her score.			
<b>Teacher Management Section</b>					

### TestMe Requirements Specification

BR_12	The teacher management section will include: <ul style="list-style-type: none"> <li>1. the examination marking</li> <li>2. the students' scores and examination time.</li> </ul>	The teacher should declare the correct answers of the exam before the exam is available to the students. This way the program will automatically evaluate the exam. In addition, the teacher can see listed all the students' scores and examination time in the teacher management page.			
BR_13	The online examination program should handle more than one exam at a time.	This said, multiple students can attend an exam at the same time without having any problems.			
<b>Website</b>					
BR_14	The website will represent the idea of the online examination program.	The website will promote and increase the downloads of the app.			
BR_15	The website should contain a login and sign up form.	Admins can login to the website and modify exam rankings, exam dates. Visitors can sign up and create an account so that they can receive updates regarding the exams.			
BR_16	The website should include a calendar with the exam dates.	For those who are not logged in/registered, only half of the calendar will be visible. Visitors have to log in/sign up in order to be able to see the full calendar. This will make sure that all the students will be informed of the exam's timetable.			
BR_17	The website will contain a review section.	The visitor can see the reviews that students have written about the program. The review section will build credibility to the visitors, so that they know			



### TestMe Requirements Specification

		our program is trustworthy. The visitor needs to have an account in order to leave a review.			
BR_18	The website should contain a gallery section.	The gallery section will contain all the types certificates our program will offer based on different subjects. The certificates will motivate students to take the exams and get higher scores.			
BR_19					
BR_20					
BR_21					
BR_22					
BR_23					

### 3.1 Non-Functional Requirements

#### 3.2.1 Product Requirements

##### 3.2.1.1 Usability

##### 3.2.1.4 Security Requirements

#### 3.1.1 User Interface Requirements

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

#### 3.1.2 Usability

Include any specific usability requirements, for example,

##### Learnability

- The user documentation and help should be complete
- The help should be context sensitive and explain how to achieve common tasks

- The system should be easy to learn  
(See <http://www.usabilitynet.org/>)

### **3.1.3 Performance**

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

- Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
- Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within a certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than "an operator shall not have to wait for the transaction to complete".

#### **3.1.3.1 Capacity**

Include measurable capacity requirements (e.g., the number of simultaneous users to be supported, the maximum simultaneous user load, per-user memory requirements, expected application throughput)

#### **3.1.3.2 Availability**

Include specific and measurable requirements for:

- Hours of operation
- Level of availability required
- Coverage for geographic areas
- Impact of downtime on users and business operations
- Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
- reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

#### **3.1.3.3 Latency**

Include explicit latency requirements, e.g., the maximum acceptable time (or average time) for a service request.

### **3.1.4 Manageability/Maintainability**

#### **3.1.4.1 Monitoring**

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

#### **3.1.4.2 Maintenance**

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

#### **3.1.4.3 Operations**

Specify any normal and special operations required by the user, including:

- periods of interactive operations and periods of unattended operations
- data processing support functions
- backup and recovery operations
- safety considerations and requirements

- disaster recovery and business resumption

### **3.1.5 System Interface/Integration**

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

#### **3.1.5.1 Network and Hardware Interfaces**

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

#### **3.1.5.2 Systems Interfaces**

Example systems interface requirements:

##### **A. System1-to-System2 Interface**

The <external party> will create and send a fixed length text file as an email attachment to [System2mail@u.washington.edu](mailto:System2mail@u.washington.edu) to be imported into the System2 system for payroll calculation. This file must be received on EDIT day by 4:00 PM in order to be processed in the EDIT night run. The requirements below document the file specifications, data transfer process, and specific schedule. This file is referred to as "FileName" in this document.

##### **File Structure and Format**

A1. The FileName file is a fixed length text file.

A2. The FileName file is an unformatted ASCII file (text-only).

A3. The FileName file contains a batch totals record and several detail records.

##### **File Description: Batch Totals Record**

A4. The batch totals record can be placed at the beginning, in the middle, or at the end of the file.

A5. The batch totals record contains the following:

- Record Type (value: XA)
- Process Type (value: A)
- Batch Number (3 digit number assigned by Payroll Dept)
- Origin Code (AIG)
- Total number of detail records
- Total deduction amount

##### **File Description: Detail Records**

A6. The FileName file contains a row for each record meeting xxx criteria.

A7. Each row in the FileName file contains the following fields, comma-delimited and encased in double-quotes where the data includes commas or spaces:

- Employee Id
- Record Type
- Process Date (MMDDYY)
- XYG Number
- Element Code
- Amount
- Amount Sign
- Year Flag

- Total Amount
- Total Amt Sign

### **3.1.6 Security**

#### **3.1.6.1 Protection**

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

- encryption
- activity logging, historical data sets
- restrictions on intermodule communications
- data integrity checks

#### **3.1.6.2 Authorization and Authentication**

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

### **3.1.7 Data Management**

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- data access rules
- data entities and relationships
- integrity constraints
- data retention
- valid range, accuracy, and/or tolerance
- units of measure
- data formats
- default or initial values

### **3.1.8 Standards Compliance**

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

### **3.1.9 Portability**

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

- Percentage of components with host-dependent code;
- Percentage of code that is host dependent;
- Use of a proven portable language;
- Use of a particular compiler or language subset;
- Use of a particular operating system;
- The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

### **3.1.10 Other Non-Functional Requirements**

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

### **3.2 Domain Requirements**

Everything related to the domain that might be needed in the project shall be mentioned in here. Sometimes the domain Requirements might be thought as part of either functional or non-functional requirements.

## **4. User Scenarios/Use Cases**

First Scenario:

Teachers log in the program and add new questions to the exam. They update older ones and delete questions they think are not suitable for the exam. They view all the questions and if they think the exam is ready they publish it.

The students log in the program and decide which exam to take. They start the exam and they complete it within the exam hours. They complete the exam and after the completion the result will appear on their screen. If the result fulfills the requirements for the certificates, they get one.

If needed the admin logs in and views teachers' and students' data as well as the students' results.

Second Scenario:

Teachers log in the program and add new questions to the exam. They update older ones and delete questions they think are not suitable for the exam. They view all the questions and if they think the exam is ready they publish it. When trying to publish it, the program shuts down and the exam cannot be published.

The students log in the program and decide which exam to take. When opening the exam, the exam appears blank and the student cannot start the exam. So the program does not work.

## APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

### Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

### Appendix B. References

List all the documents and other materials referenced in this document.

### Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview

### TestMe Requirements Specification

BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

## **Appendix D. Organizing the Requirements**

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

### **By System Mode**

Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

### **By User Class**

Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

### **By Objects**

Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

### **By Feature**

A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

### **By Stimulus**

Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

### **By Response**

Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

### **By Functional Hierarchy**

When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

### **Additional Comments**

Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.