Interactive Requirements Exercise

SQGNE Software Quality Group of New England

Steve Rakitin

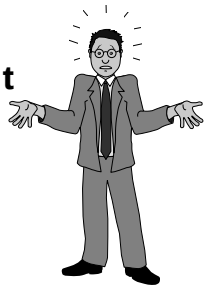
Howie Dow

Tonight's Topic

- **Requirements**
- **Alternatives to English**
- **Exercise**

Why are requirements hard to write?

- Requirements are written in English
- English is inherently vague and imprecise
- We rarely train people in how to write good requirements
- We often have trouble separating requirements (WHATs) from design (HOWs)
- Impact of poorly written or non-existent requirements not understood...
- Misconception - spending time writing requirements delays product release...



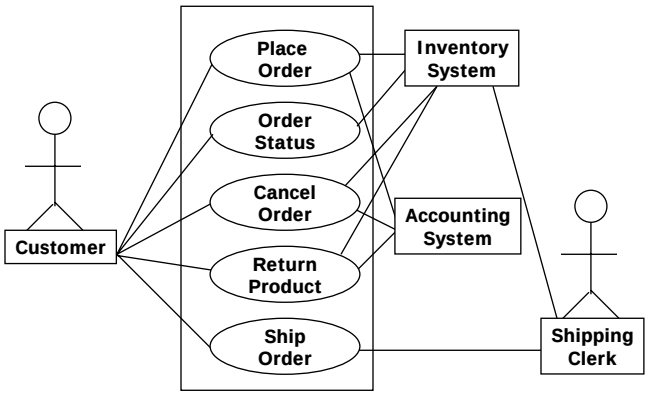
Writing Requirements

- Alternative Techniques
  - reduce ambiguity by expressing requirements in a manner that leads to better understanding, a more coherent design, and more effective testing...
- Some examples:
  - Use Case Diagrams
  - Flowcharts
  - Structured English
  - Truth Tables
  - State Transition Diagrams
  - E-R Diagrams
- Excellent tools for expressing requirements in ways that lead to clear understanding

Write Testable Requirements

- **Use Case Diagrams**

  Express requirements in ways that facilitate understanding

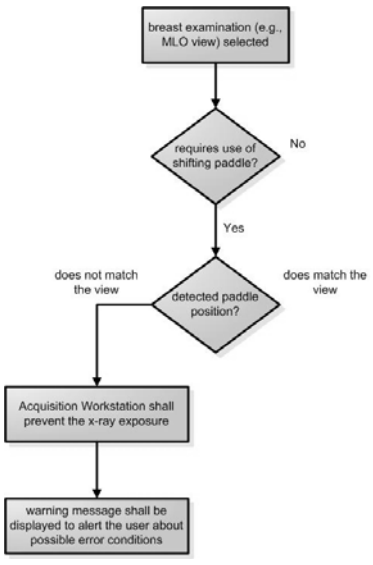


Schneider, G. and Winters, J., Applying Use Cases: A Practical Guide, Addison-Wesley, 1998.

Write Testable Requirements

- **Flowcharts**

When a breast examination is selected requiring the use of a shifting paddle, if the detected paddle position does not match the view, the Acquisition Workstation shall prevent the x-ray exposure until the faulty condition is corrected. A warning message shall be displayed to alert the user about possible error conditions.



Write Testable Requirements

- **Structured English**

  When running the calculation rule, if a numerical operation is to be performed such as multiplication or division, the system shall only run the rule if value to be used is in numerical form.

- **Re-write using Structured English**

```
IF      numerical operation is to be performed when running
        calculation rule
THEN    IF      value to be used is in numerical form
        THEN    run calculation rule
        ELSE    ??
ELSE    ??
ENDIF
```


Write Testable Requirements

- **Truth Tables**

Change User Password Example OP = Old Password NP = New Password

User enters new password. Application confirms if new password meets the following configuration rules.

1. **If OP is correct, NP and Confirm NP are same, pass configuration edit, and has not been used during prior two changes, confirm change with a successfully changed password message. An "OK" button brings user to Home page. Error Message = Password successfully changed.**
2. **If OP is correct and NP and Confirm NP match but do not conform to configuration settings, a message describing error is displayed. OP, NP and Confirm NP are blank after user selects "OK" on error message. "OK" button brings user to Change Password screen. Error Message = Password you entered does not conform to format specified by your system administrator. Enter a valid password.**
3. **If OP is valid and NP and Confirm NP do not match, a message describing error is displayed. OP, NP and Confirm NP are blank after user selects "OK" on error message. "OK" button brings user to Change Password screen. Error Message = NP and Confirm NP entries do not match. Try again.**
4. **If OP is correct and NP and Confirm NP match and pass configuration but has been used prior by user during previous two changes a message is displayed. OP, NP and Confirm NP are blank after user selects "OK" on error message. "Ok" button brings user to Change Password screen. Error Message – Password has been used too recently. Try again.**
5. **If NP and Confirm NP match and pass configuration but OP is invalid, a message is displayed. OP, NP and Confirm NP are blank after user selects "OK" on error message. "Ok" button brings user to Change Password screen. Error Message – OP entered is invalid. Try again.**

Write Testable Requirements

- **Truth Tables**

OP Confirmed	NP and Confirm NP match	Password Rules followed	NP not used in last two changes	Password change successful?	Display Message
TRUE	TRUE	TRUE	TRUE	Yes	1
TRUE	TRUE	FALSE	DON'T CARE	No	2
TRUE	FALSE	DON'T CARE	DON'T CARE	No	3
TRUE	TRUE	TRUE	FALSE	No	4
FALSE	DON'T CARE	DON'T CARE	DON'T CARE	No	5

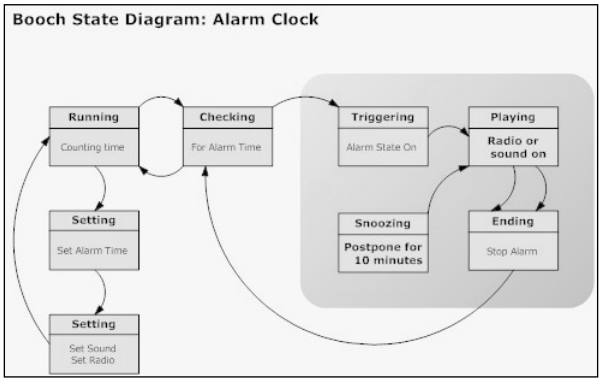
Messages:

1. "Password successfully changed"
2. "The password entered does not conform to the format specified by your sys admin. Enter a valid password."
3. "New Password and Confirm New Password entries do not match. Try again."
4. "The password you entered has been used recently. Try again."
5. "The Old Password you entered is invalid. Try again."

How many tests are required to test these requirements?

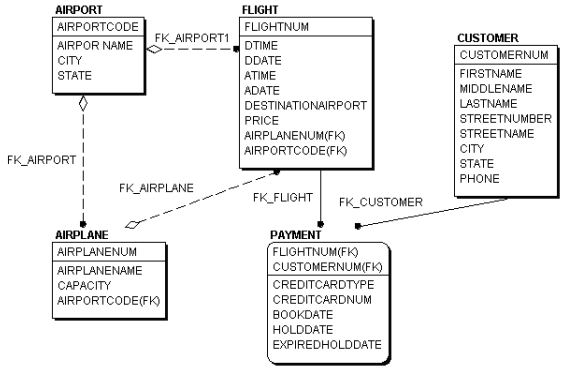
Write Testable Requirements

- **State Transition Diagrams**
  - **Essential for state-driven and real-time systems**



Write Testable Requirements

- **E-R Diagrams**
  - **Essential for database-centric systems**

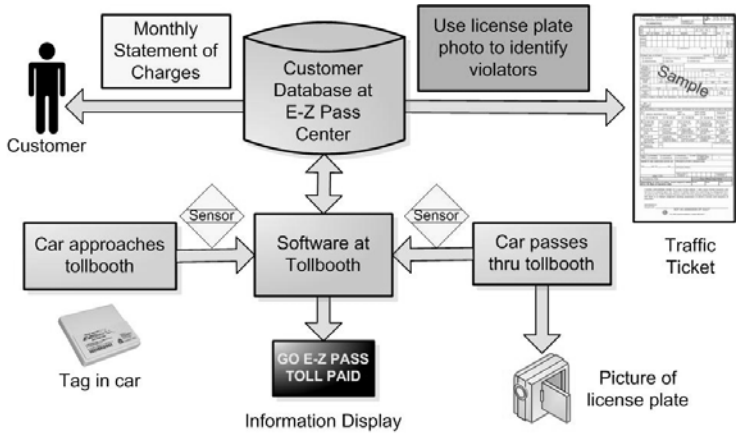


Exercise

- **Break up into groups**
- **Each group select an alternative technique**
- **Write requirements using that technique**
- **For the technique chosen, pay attention to:**
  - Strengths
  - Weaknesses
  - Issues or obstacles
  - What tools might help
  - Anything else that comes to mind
- **In 30 minutes (or less) we will share results**

Exercise

Exercise



Assumptions for Requirements Exercise

1. Assume only passenger cars.
2. Assume drivers have valid E-Z Pass tags and have set up a valid account with E-Z Pass Center.
3. Assume there is no gate as shown in the picture...
4. Assume that toll to be paid is determined by distance driven - there is a tollbooth where drivers enter and another tollbooth where they exit.
5. Focus only on software that resides at tollbooths – not at central location.

Reading List

- Wiegers, K. E., Software Requirements, Microsoft Press, 1999
- Wiegers, K. E., More About Software Requirements – Thorny Issues and Practical Advice, Microsoft Press, 2006
- Goldsmith, R., Discovering the REAL Business Requirements for Software Project Success, Artech House, 2004
- Gause, D. and Weingberg, G., Exploring Requirements: Quality Before Design, Dorset House, 1989.
- Sommerville, Ian, and Pete Sawyer, Requirements Engineering: A Good Practice Guide, John Wiley & Sons, 1997
- Robertson, S., "Requirements and the Business Case", *IEEE Software*, Sept-Oct 2004, pp. 93-95

Thank you…

- **If you have questions, please call or e-mail…**
- **Subscribe to my e-newsletter…**

- **For a free subscription and to view past newsletters, visit www.swqual.com**

Software Quality Consulting Inc.

Steven R. Rakitin
President

- Consulting
- Training
- Auditing

Phone: 508.529.4282
Fax: 508.529.7799

www.swqual.com
steve@swqual.com

SQGNE Software Quality Group of New England

SQGNE is made possible by the support of our sponsors:

SOFTWARE QUALITY CONSULTING
consulting • training • auditing

Welcome to our 16th season!

- An all-volunteer group with no membership dues!
- Supported entirely by our sponsors…
- Over 700+ members
- Monthly meetings - Sept to July on 2nd Wed of month
- E-mail list - contact John Pustaver **pustaver@ieee.org**
- **NEW SQGNE Web site: www. sqgne.org**

Volunteers / Hosts / Mission

Volunteers

- John Pustaver - Founder and Director
- Steve Rakitin – Programs and web site
- Gene Freyberger – Annual Survey
- Dawn Wu – our new greeter!!

Our gracious Hosts

- Paul Ratty - room, copies, cookies
- Tom Arakel - room, copies, cookies
- Margaret Shinkle - room, copies, cookies
- Jack Guilderson – A/V equipment

Mission

- To promote use of engineering and management techniques that lead to delivery of high quality software
- To disseminate concepts and techniques related to software quality engineering and software engineering process
- To provide a forum for discussion of concepts and techniques related to software quality engineering and the software engineering process
- To provide networking opportunities for software quality professionals

ASQ Software Division

- Software Quality Live - for ASQ SW Div members…
- Software Quality Professional Journal www.asq.org/pub/sqp/
- CSQE Certification info at www.asq.org/software/getcertified
- SW Div info at www.asq.org/software

SQGNE 2009-10 Schedule

Speaker	Company/Affiliation	Date	Topic
Eric Lotter	Surgient	9/9/09	Using Virtualization to Accelerate Quality/Test Cycles
Steve Rakitin	Software Quality Consulting	10/14/09	Software Quality Assurance Turns 50 A Critical Look at the Profession
Howie Dow and Steve Rakitin		11/11/09	Interactive Requirements Exercise…
Michael Mah	QSM Associates	12/9/09	Rightsizing Your Project in a Down Economy
Robin Goldsmith	GoPro Management	1/13/10	I went to a Testing Conference and all they talked about was Requirements
Stan Wrobel	CSC	2/10/10	To be announced…
Billie Bell	Intuit	3/10/10	End-to-End Testing in a SaaS environment: Extending the Definition of Quality
Linda McInnis		4/14/10	Metrics: The Where, How and Why?
Urvashi Tyagi	Microsoft	5/12/10	A day in the life of a tester at Microsoft…
Brian LeSuer	Star Quality	6/9/10	To be announced…
Everyone		7/14/10	Annual Hot Topics Night…

Tonight's Speaker…

Interactive Requirements Exercise
Howie Dow and Steve Rakitin

Requirements are the most important information a software project team needs. Developers and testers all depend on clear, unambiguous requirements to do their work. Yet, few projects have such requirements. Most requirements are written in English. Good requirements are hard to write because English is an inherently vague and ambiguous language and because we don't train people in how to write good requirements using English. In this talk, we will review several graphical alternatives to English that are much less ambiguous. Small groups will work on a expressing requirements using different techniques so that we can see first hand the benefits of graphical techniques…