

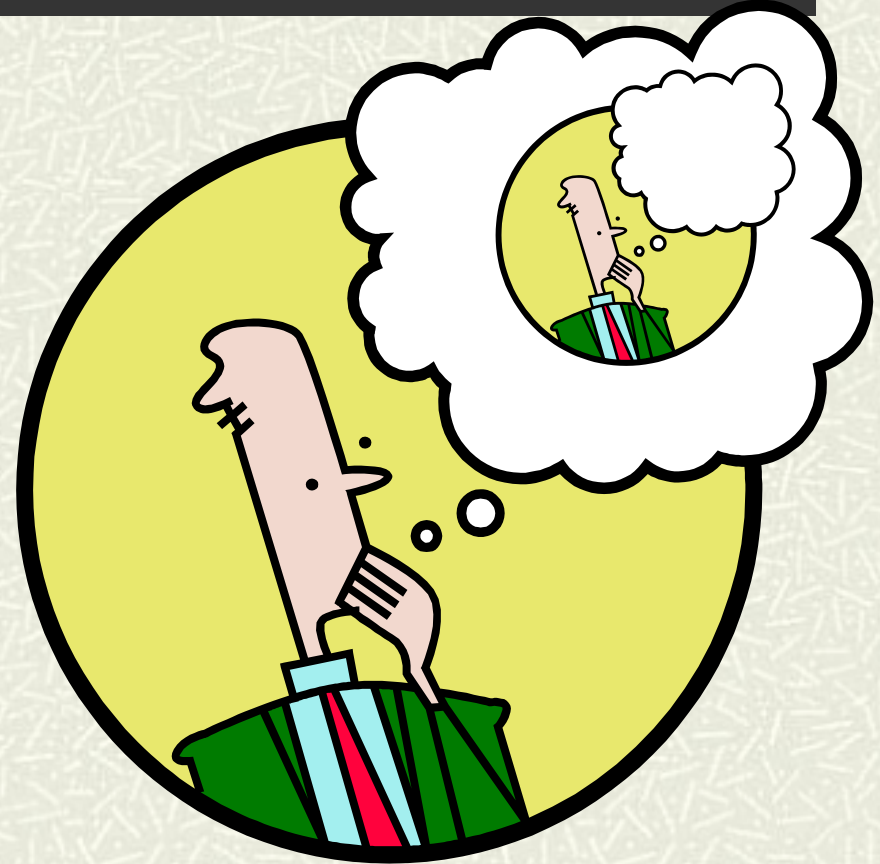
Insight into reusable test cases using Questioning Patterns

Agenda

- # Test case reuse: What, Why, and How
- # Q-Patterns: What, Why, and Where
- # Identifying Q-Patterns
- # Tailoring/Authoring Q-Patterns
- # Using Q-Patterns in testing
- # Other uses of Q-Patterns
- # Possible issues with Q-Patterns and their resolution
- # Where to find more information

Deja Vu

- # Have you ever looked at a new system to be tested and thought “I have done this before”?
- # What are the things that you had “done before”?



Learning and communicating it

- # How do you utilize your learning in new situation?
- # How do you communicate your learning?
- # How do you learn from other's learning?

If wishes were horses...

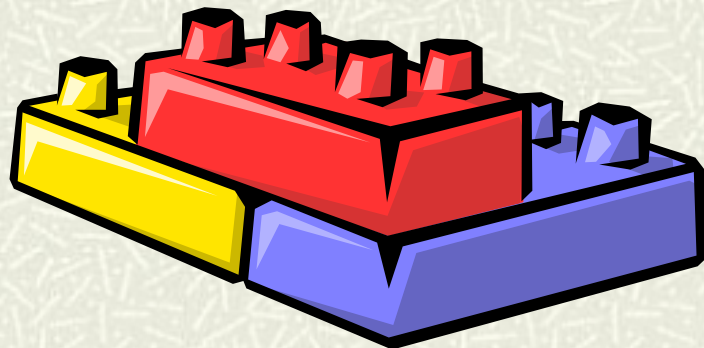
Have you ever wished

- more domain specific testing information
- more platform specific testing information
- More technology specific testing information
- more use-cases for the given type of solutions
- Pre-written test cases that you could use for testing or for learning
 - Have you looked at the test cases for the famous triangle problem?

One answer to all the wishes-Reuse

Test case reuse: What is it?

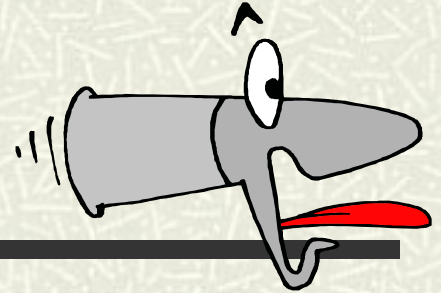
- # Write once, use multiple times
 - For similar applications
 - For different applications with similar features
 - For the same or different organization
- # Does not imply 100% reuse
- # Customization of tests may be necessary



Test case reuse: Issues

- # Specs reuse = test case reuse?
- # Changes of varying granularity may be required to make test cases reusable?
- # Lack of an accepted format for writing test cases for reuse
- # Lack of organized sources for these tests
 - Information on the net is too little, too scattered
- # IP issues: Organizations not too keen to share tests.
- # Q-Patterns solve some of these issues

Q-Patterns: A Silver Bullet?



Q-Patterns provide:

- Means to communicate experience
- Aid in writing test cases
- Provide a mechanism for test case reuse
- Also aid in reviewing specifications and design

What are Q-Patterns

- # A set of interrelated QUESTIONS grouped together
 - relate to some aspect of user or software requirements
 - provide various alternatives to arrive at a solution.
- # You can also think of them as
 - Set of pre-cooked test cases
 - A tool for requirement elicitation, defining your specs etc.

Q-Pattern: An example...

- # Consider this requirements fragment for login functionality of a system
 - Present a screen for login. The user id field is a combo box and password field is a text field.
 - Successful login takes you to the ‘next’ screen
 - Unsuccessful login results in message to the user about login failure

Some Questions...

User ID/Password constraints/properties

- What is the min/max length of user id/password?
- Are they mandatory to be alphanumeric/numeric only?
- Is user id/password case sensitive?

UI

- Is the field wide enough to display user id?

Login failure

- What is the number of allowed retries for login failure?
- What happens after all retries are exhausted?
- Where is cursor placed after login failure and what fields are blanked, if any?

More Questions

Does the password expire?

- In how many days? (Say X)
- Can the user set the value of X?
- Can administrator set “Password does not expire” for a user?

Password change

- Display in clear text or as some other character as * etc.? Confirm password feature?
- When shown as *, how many characters are displayed?
- When the password is changed after expiry can it be reset to the same password again? Or last N number of password can not be used?
- Can it be reset to the same password again?
... (And many more questions)

Was that a Q-Pattern?

- # We saw unstructured questions
- # We saw incomplete questions
- # Need to put more thoughts into a well defined structure to create a Q-Pattern
 - Why is a structure required?
 - What structure can exist for Q-Patterns of
 - 1. Different granularity
 - 2. Different domains
 - 3. Different concepts etc.

Structure of a Q-Pattern

- # Name of the pattern
- # Intent/Explanation/definition
- # Questions related to:
 - Administration
 - Usage
 - User Interface
 - Internationalization
 - Localization
 - Security
 - Performance
 - Response Time (Fetch, insert/delete/update, Display)
 - Concurrency
 - Max and Min parameters (Data size etc.)
 - Memory requirement
 - Disk Space
 - ...
- # Examples
- # Associated patterns
- # Specialization

Q-Patterns: An example

NAME

- # Password based Login Management

INTENT

- # Enumerating various strategies for password based authentication of a system or a user where the mechanism involves providing a character based user id and password

Administration

- # Does the system have a distinction of user and administrator?
- # What is the default password for administrator?
- # Are multiple administrators allowed?
 - Who creates administrative users?
 - Is there any super admin-admin hierarchy?
 - What is the difference of rights between users and administrators?
- # User Creation
 - Who can create users?
 - What is the default value for password of new users?
 - Can users be created in bulk by importing from lists, files etc.
- # Password resetting
 - Can an administrator read the password of a user?
 - Can administrator reset the password for other users?
 - Can administrator's password be reset?

Usage

- # User ID/Password constraints/properties
 - What is the min/max length of user id/password?
 - Are they mandatory to be alphanumeric/numeric only?
 - Is user id/password case sensitive?
 - Are Blank password allowed?
 - Can characters like #,\$,or accented characters such as ç,è be used?
- # Successful Login
 - Is the next screen presented to the user based on user role?
- # Is there a 'Save Password' feature?
 - Can this be turned off?
 - Can saved passwords be "unsaved"
 - What is the limit on number of saved used id-password combinations?
 - Where is this information saved? (Also see section on Security)

Usage

- # Login failure
 - What is the number of allowed retries for login failure?
 - What happens after all retries are exhausted?
- # Does the password expire?
 - In how many days? (Say X)
 - Can the user set the value of X?
 - Can administrator set “Password does not expire” for a user?
- # Password change
 - How is password change affected?
 - original password required?
 - Is Confirm password used?
 - Display in clear text or as some other character as * ?
 - When shown as *, how many characters are displayed?
 - Does the password expire?
 - When the password is changed after expiry can it be reset to the same password again? Or last N number of password can not be used?
 - Can it be reset to the same password again?
- # Can user request a password reset, if he has forgotten the password?

UI

Login screen

- When the application is invoked, is the cursor places at the right place say, user id field
- Is the field wide enough to display user id?
- Is password shown as stars while entering or changing/resetting the password
- How many stars are shown
 - When password is being entered?
 - When it is to be changed?
- Is the tab order as desired?
- Is enter the default button?
- Is user id field a text field or a combo box? Can user be selected instead of being entered?
- Are shortcuts and hotkeys present?

What is the strategy for conveying login failure?

- Exact error cause (user does not exist, password is incorrect) shown or a generic message about any one of them being incorrect shown?
- Where is cursor placed after login failure message is dismissed?
- User id/password or both blanked out after incorrect login?
- What message is conveyed if all retries are exhausted? Can a legitimate user get help from that message?

Is the feature accessible?

Security

- # Where are the passwords stored?
- # Are they encrypted before storing? If yes what is the encryption algorithm used?
- # Encryption before transmission?
- # Number of failed login allowed ?
- # Expiry time for the password?
- # Display of password in debug logs?
- # Expiry of session after some interval of inactivity?
- # Where does save password feature store the password? Is the password encrypted? What is the algorithm used?

Performance

Response time performance

- What is the time taken for authentication? (It may depend upon...)
 - Repository for password storage
 - Encryption algorithm
 - Connectivity
- What is the time taken for displaying next screen?
- What is the time taken for notification of login failure?
- What is the response time for above scenarios when there are multiple simultaneous logins happening?

Storage performance

- Is the storage independent of user id and password length?
- Does the storage adjust itself when users are deleted?

Examples

- # OS login
- # Web based mail systems
- # Console based login (Telnet)

Associated patterns

- # Combo Box
- # Error Handling
- # Access Rights

Specialization

- # Prune the questions to suite your needs
- # Add more questions to the specialized list OR enhance the parent Q-Pattern.
- # Example: Password based Login Management System for Web based software. (Specialize for say, use of GET/POST methods, Security in transmission etc.)

Combo Box Q-Pattern

- # Name: Combo Box
- # Intent: Behavior of combo boxes
- # Administration: NA
- # Usage:
 - Where does the combo-box get its values from?
 - What if the data source fails?
 - Are values Sorted?
 - How are new values added? Deleted?
 - Where are new values displayed? Beginning, end or sorting based position?
 - Is it single or multi select? How is multiple selection done?
 - What font is used? Does it change with system settings?
 - Are text adornments such as bold, italics or underline allowed?
 - Are there any actions associated with tab movement, mouse clicks?

Combo Box Q-Pattern

User Interface

- Is box wide enough to display the widest element? If not, is scrolling possible?
- How is selection displayed? (Bounding by dashed line or highlighting or checkboxes etc.)
- If highlight is used, is the contrast of font and highlight color easy to read?
- Does the box drop down to reveal all entries or some? How do you scroll vertically?
- Can the box drop down and selections made using keyboard? Using Mouse?
- What are the Background/Foreground colors? Can they be changed by the user?
- How is text aligned?
- Is there type-ahead feature?
- Is the selection retained?

Internationalization

- Does the box accept and display extended ASCII characters?
- Does the box accept and display multi-byte characters?
- Does the box accept and display right-to-left (Arabic, Hebrew...) scripts?

Combo Box Q-Pattern

- # Security: NA
- # Performance
 - Time taken to populate the combo box
 - Time taken for the drop down action
 - Time taken for selecting items
 - Time taken for adding/deleting items
- # Examples: File Open dialog (File name)
- # Associated patterns: None
- # Specialization: Depending on toolkit used

Error Message Q-Pattern

- # Name: Error Handling
- # Intent: Handling and Display of errors.
- # Administration
 - Can a user/administrator decide not to display errors?
 - How can the errors be displayed again?
 - Can the errors be also logged in some files?
- # Usage
 - Are errors categorized? For example, fatal, serious, minor etc.?
 - Can error messages be modified or customized?
 - Is the error message text useful? Does it explain the problem
 - Does the error message suggest possible solution or sources for further help?
 - Does the error message give you a facility to send the error report to its source?
 - Is user consent sought before any error information is sent anywhere?
 - Does the feature include collection of all relevant information from various logs and configuration files?

Error Message Q-Pattern

User Interface

- Are error messages displayed as dialogs, part of a screen say, status bar or logged into text logs?
- Are error messages differentiated from warnings and information messages?
- How is this differentiation done?
- When action is required on errors what kind of options are provided to the user?
- When error dialogs are dismissed, where does the control go?

Internationalization

- Do error messages support extended ASCII and multi-byte character set?

Localization

- Are error messages localized?
- Can the language of the error messages be changed?

Error Message Q-Pattern

- # Security: NA
- # Performance: NA
- # Accessibility: Are the errors accessible to challenged people?
- # Examples: Far too many 😊
- # Associated patterns: NA
- # Specialization: Depending on the application requirements

Looking back

- # Q-Patterns are general questions related to some aspects of requirements
- # Q-Patterns can provide a web of interrelated patterns which can give you lot of test cases.
- # Q-Patterns can handle various problems of various granularities
- # The real value of Q-Patterns can be realized when lot of Q-Patterns are available easily
- # These record the condensed experience of a lot of people and thus can serve as training fodder for people fresh to testing

Q-Patterns and Columbus

- # Q-Patterns are tools for Exploration
- # These can help exploratory testing: By suggesting alternate paths
- # These can track the paths explored: By recording newer questions, by weeding out questions not relevant to the application

Going Forward

- # Identifying Q-Patterns in organizational context
- # Understanding how they can be used with an example
- # Possible issues with Q-Patterns and their possible resolution
- # New things that need to be done

Defining the Organizational Context:

"situation" specific to an organization. The situation may be processes, policies, development/testing methods, strategies etc.

Identifying Q-Patterns

- # Need to understand the organizational context to write new Q-Patterns or to identify the Q-Patterns applicable to the given organization
- # Once identified, we can tailor them

Process of using Q-Patterns

- # Identify the candidates Q-Pattern
- # Search for a existing similar Q-Pattern
- # If available, tailor it otherwise write it
- # Test it by applying at some relevant places.
- # (Optional) Submit the Q-Pattern to the repository

Identifying candidates Q-Patterns

Requires

- Good understanding of the context
- familiarity with the domain
- familiarity with the requirements
- familiarity with the implementation

Identify repetition

- Within the same software
- Across software

Example: Access rights, Password management, List handling...

Identifying repetition

- # Use requirements or functional specs to identify the similarities
- # Some common heads
 - Security
 - Performance
 - User Interface
- # Existing test cases
- # Defect reports
- # Defects found in exploratory testing

Applying Q-Patterns

- # Q-Patterns can be applied as a WEB of interrelated patterns.
- # For this, Identify Q-Patterns for the
 - Framework /architecture/Domain <Coarse grained>
 - Individual components
 - Sub-components
 - Features and functions
 - Individual elements, Algorithms, UI screens etc. <Fine grained Q-Patterns>

An Example

Let us consider a software with client-server architecture providing:

- # Logon (As administrator/end user) and logoff.
- # Administrator can
 - a. Add/delete/modify users from list of users.
 - b. Search for a user on the basis of name/user id
- # End user can create expense records that are submitted to the database.

Candidate Q-Patterns

Architectural/Domain Q-Patterns:

- Client-Server or Web Based architecture.
- Banking, Insurance, Medical, Mobile computing, e-commerce, m-commerce etc.

Individual components:

- Client.
 - Browser client: Browser specific Q-Patterns OR
 - Application on a machine: <Q-Pattern, if any>
- Server: <Implementation based Q-Patterns, if any>

Sub-Component based Q-Patterns, if any

Candidate Q-Patterns

Feature based Q-Patterns

- Logon
- Access rights <this one brings out a fact that we have missed assigning users a role admin OR end user>
- Database Transactions

Individual elements/Algorithms etc.

- Search
- Sorting
- List
- Q-Patterns for individual UI elements <Edit, Combo/List box etc. .

Q-Patterns: beyond the sales pitch

- # What are the caveats?
- # What are the possible improvements?

Possible Issues

- # Possibility of
 - unwieldy collection of questions.
 - duplicate questions in various related/unrelated patterns
- # All the sections described in the proposed structure may not be universally applicable
- # Lack of a repository of Q-Patterns
- # IP issues

Possible Improvements

- # Versioning the patterns
 - Required for sharing the patterns unambiguously
- # Some classification of Q-Patterns as Organizational, Design, System Testing Q-Patterns
- # Use of OO techniques to take care of similarities and specializations
- # A mechanism (UML based?) for representing the web of Q-Patterns for a robust and intuitive structure for referencing other Q-Patterns

Searching for Q-Patterns

- # www.WhatIsTesting.com will be used as a repository of Q-Patterns.
- # We will try to create an easy-to-search and complete repository of Q- Patterns and related links
- # You may search the repository for existing Q-Patterns before you proceed to write your own

Testing/Submitting the Q-Pattern

- # Use your Q-Pattern in some known situations and see if these yield you a sufficient quantity of good quality test cases. If it does then probably it will yield you good test cases in new situations
- # To share it with the tester community submit it on www.WhatIsTesting.com.
- # It will be reviewed by others and put in the repository with full credit to the author(s).

More information

Questioningpatterns group at yahoo groups

<http://groups.yahoo.com/group/questioningpatterns/>

<http://www.WhatIsTesting.com>

[Mail: vipul at WhatIsTesting.com](mailto:vipul@WhatIsTesting.com)



Thanks

