

# Applying modern software development techniques to UI testing

# Ultimate Software: Mission Statement

To provide United States and Canadian businesses with 200 or more employees the highest quality, most complete, and well-integrated suite of strategic human resources, payroll, and talent management solutions.

## Who we Are

Michael Longin – Ultimate Software  
Software Engineer  
Certified Scrum Master  
Project Lead - SWAT

Christopher Taylor – Ultimate Software  
QA Automation/Tester  
Certified Scrum Master

# Tools we will use

- SWAT
  - Simple Web Automation Toolkit
  - <http://ulti-swat.wiki.sourceforge.net/>
- Fitnessse
  - Wiki based requirements test tool
  - [www.fitnessse.org](http://www.fitnessse.org)
- Both are open source and can be used either together or stand alone

## Goals for this session

- Attendees should be able to apply modern software development techniques to UI testing to create more agile and resilient tests and improve overall quality.
- “*Automated testing* done right is Software Development”
  - Elfriede Dustin and Marcus Borch @ GTAC 2008

# Syllabus

- Why we do UI testing
- What is wrong with the record\playback technique
- Introduction to example project
- Applying Test First Development
- Refactoring
- Pair testing
- Compare our way vs. the “old way”

# Why we do UI testing

- All the way through
  - Through the product
  - Through the layers
  - Its truly how the user will use your system
  - 100% Functional

# Record-Playback technique failures

- **Brittle**
  - Easily broken when features change, even when the feature is not part of the test
- Very hard to update without re-recording
- **Not Agile**
  - Can only be done when code is complete
- Only tests what you record and not what you expect
- Time consuming because of the above
- Hard to read and understand

# Our User

- Name: JB
- Occupation: Midlevel government employee
- Skills:
  - Web browsing
  - Word
  - Excel



# User Story #1 : Login

- As a user I want to provide my username and password to login
- Conditions of Satisfaction
  - Login and Password boxes are present
  - Title should read “M&C Life Insurance”
  - Valid Login should lead to homepage
    - Example Login: jbauer\password
  - Homepage should read “Welcome”

## User Story 2: View My Name

- As a user I want to be welcomed by my name after I login
- Conditions of Satisfaction
  - My first and last name are correct
  - Title should read “Welcome ‘first name’ ‘last name’”
    - Example: Welcome Jack Bauer

## User Story 3: Change My Name

- As a user I want to be able to update my name
- Conditions of Satisfaction
  - I should get to the page by a menu located on the home page
  - There should be a first name and last name text box on the screen
  - Pressing save should bring me back to the homepage

# Applying TFD

- Test can be written before the feature
- Allows parallel effort of development and automation
- Turns vertical slicing into completed automation
- Starts with well written conditions of satisfaction
- Consistent naming standards are key
  - txb -> Textbox
  - btn -> Button
  - lbl -> Label
  - \*if naming is wrong, can be easily updated
- If the developer writes the automation, becomes almost a UI TDD

- Example User Story #1
  - Title should read “M&C Life Insurance”
    - Title means an H2
    - Text reads: “M&C Life Insurance”
  - Login and Password boxes are present
    - 2 text boxes
    - Login - >txbLogin
    - Password -> txbPassword
    - Login Button -> btnLogin

# Applying TFD continued

- Example Automation Code

```
|AssertElementExists|expression|innerHTML:M.C Life Insurance|h2| | |
|SetElementAttribute|Id|txbLogin|value|jbauer|input|  
|SetElementAttribute|Id|txbPassword|value|password|input|  
|StimulateElement|Id|btnLogin|onclick|input|  
|AssertElementExists|Expression|innerHTML:welcome|
```

# TFD Demo

- Demo:

# Refactoring

- Why refactor
  - Less time to write automation
  - Less maintenance
  - Creates reusable code
  - Creates a domain specific language
- What is refactoring
  - For our purposes
    - Turn frequently used blocks of code into single line reusable entities
    - Use variables to make tests more robust
    - Creates an easily readable test by using English named functions

## Refactoring continued

- Allows tests to be easily updated if core features change
  - If login changed for example, only one update needed
- Allows those beginning to write tests to take advantage of previously created work
- Makes tests much easier to read and debug

# Refactoring continued

- Example:

```
|OpenBrowser|
```

```
|NavigateBrowser|www.m&cinsurance.com|
```

```
|SetElementAttribute|Id|txbLogin|value|${userName}|input|
```

```
|SetElementAttribute|Id|txbPassword|value|${password}|input|
```

```
|StimulateElement|Id|btnLogin|onclick|input|
```

- Becomes:

```
!define loginUserName (jbauer)
```

```
!define loginPassword (password)
```

```
!include .Macros.Login
```

- Example:

```
variable defined: loginUserName=aut  
variable defined: loginPassword=password
```

```
▶ Included page: .SwatMacros LoginToServer
```

# Refactoring Demo

- Demo:

# Pair Testing

- Benefits rival those of Pair Programming
  - [http://en.wikipedia.org/wiki/Pair\\_programming#Benefits](http://en.wikipedia.org/wiki/Pair_programming#Benefits)
    - (Yes we did infact just site wikipedia)
- Can be very effective for both writing automation and locating missing requirements
- Two heads are always better then one
- While automating you can also accomplish exploratory testing
- Can replace manual testing phase with the creation of automation

## Compare 2nd test example vs a “recorded version of the same test”

- Unreadable
- Unorganized
- Not useable as documentation (concept of testable documentation)

# Where to get help

- Questions
  - SWAT:
    - [http://sourceforge.net/forum/?group\\_id=199701](http://sourceforge.net/forum/?group_id=199701)
  - Fitnessse:
    - <http://tech.groups.yahoo.com/group/fitnessse/?v=1&t=search&ch=web&pub=groups&sec=group&slk=2>
  - Email
    - [michael\\_longin@ultimatesoftware.com](mailto:michael_longin@ultimatesoftware.com)
    - [christopher\\_taylor@ultimatesoftware.com](mailto:christopher_taylor@ultimatesoftware.com)
  - Blogs
    - [devXero.wordpress.com](http://devXero.wordpress.com)
    - [www.agile-tester.com](http://www.agile-tester.com)
- Websites
  - <http://www.fitnessse.org/>
  - <http://ulti-swat.wiki.sourceforge.net/>

# Questions

- Questions????



In cooperation with Great Place to Work® Institute and Society for Human Resource Management.



Powered by Great Place to Work® Institute and Society for Human Resource Management.



Powered by Great Place to Work® Institute and Society for Human Resource Management.



Powered by Great Place to Work® Institute and Society for Human Resource Management.