Test Tool – Make or Buy?

Stan Wrobel

Agenda

- Introduction
- A Tale of Four Test Tools
  - Exescript – the shell script on steroids
  - Raging Bull – web wonder
  - UFIT – the stealth tool
  - TTT - just test everything!
- Pros/Cons
  - Buying a Tool
  - Building your own Tool
- Questions

RTR – the Middleware Contender

- **Product under Test:**
  - **Reliable Transaction Router – Fault-tolerant TP Middleware**
    - **C API – 15 elements – transaction-based**
    - **Runs on 5 Unix flavors, Windows and VMS**
    - **Fault-tolerant distributed network architecture**
    - **Easily scriptable command line interface**
    - **Key Feature is Transaction integrity – used by banks and stock exchanges**
    - **Minimal budget allocation to testing**

RTR

- Make or Buy?

?

Execscript – shell script on steroids

- **Execscript features**
  - **Multi-platform C application for managing and executing multi-script scenarios**
  - **Time-slipping via primitive substitution mechanism**
  - **Result comparison against known-good result using substitution to filter out the chaff**
  - **Automatic collection of results with email attached to nightly build email**

Execscript – Cost/Benefit

- Cost was very low
  - Specification was 5 pages
  - Coding/debugging took roughly one man-month
  - Much more time spent developing/tailoring scripts and building scenarios
  - Documentation and training were minimal
  - Used in-house C programming and shell/DCL scripting expertise
- ROI was high
  - Basic test scenario served as automated smoke test and was integrated into nightly software build scripts
  - More complex scenarios served as regression tests. Regression test suite was gradually augmented as new scripts based on bug reports were developed and were used to verify bug fixes
  - Developing new scripts required minimal skill set

RagingBull.com – Investor Empowerment

- **Product under Test:**
  - **Ragingbull.com – Investor Message Boards**
    - **Single database server, 8 web servers**
    - **20 million page views per day**
    - **Real-time quotes and News articles**
    - **Magnet for spammers, hackers, pump-and-dumpers**
    - **User account and email options**
    - **Shoestring budget based on ad revenue**

RagingBull.com

- Make or Buy?

?

Astra Site Manager – freeware option

- **Site Manager features**
  - **Web page link-checker**

Astra Site Manager – Cost/Benefit

- Cost was free
  - Freeware
  - Easy to use
- ROI was nevertheless low
  - Only did part of what testers needed done
  - No Load testing
  - Pages were still verified by eyeballs

ETMS – prototypical dilemma

- **Product under Test:**
  - **Enhanced Traffic Management System**
    - **X Windows graphical user interface**
    - **Flight Plan and real-time radar inputs**
    - **Tracking of planned load on airports and air spaces hours in advance**
    - **Planning, Modeling, Implementation of Ground Delay and Reroute initiatives**
    - **New releases every 6 months**

ETMS

- Make or Buy?

?

UFIT– Universal Flight Injection Tool

EXPERIENCE. RESULTS.

- **UFIT features**
  - **Execution of short scripts via a custom command line interface**
  - **Time-slipping via sophisticated substitution mechanism**
  - **Result comparison against known-good result using substitution to filter out the chaff**
  - **Built primarily to test the Airline interface into ETMS, it also was capable of testing flight plan processing and system management commands**

UFIT Time-Slipping

EXPERIENCE. RESULTS.

- **4.2.1 Specifying Dates and Times**
- While the times of the messages contained within a sequence file can be specified as a definitive integer, UFIT provides a way to specify variable dates and times that will be updated at run time. This means that each time you run a sequence file using UFIT, you will NOT have to change the dates and times.
- All variable dates and times are based on the current date and time that is saved at the start of the test run. On the first pass through the test sequence file, all date and time variables are replaced with actual timestamps.
- For each time variable, the word before the plus '+' sign tells UFIT how to format the time. The numbers after the plus sign tells UFIT how many hours and minutes to add to the timestamp saved at the beginning of the test run.
- For example, UFIT replaces the variable [Time+1:5] with 1305 if the time at the start of the test was 12:00. Likewise, UFIT replaces [MonthDayTime+1:15] with 04141315 if the current date and time is April 14, 12:00. Adding 24 hours to a variable causes UFIT to replace the variable with tomorrow's date. For example, UFIT replaces [MonthDay+24:0] with 0415 if the current date is April 14 and it replaces [JulianDate+24:0] with 2414 if the current Julian date is 2413. Any number of hours and minutes added to the test run time that would cause the time to be greater than 23:59 will increment the date. So, UFIT replaces [MonthDay+2:0] with 0415 if the date and time at the start of the test run was April 14, 23:00.

UFIT – Cost/Benefit

EXPERIENCE. RESULTS.

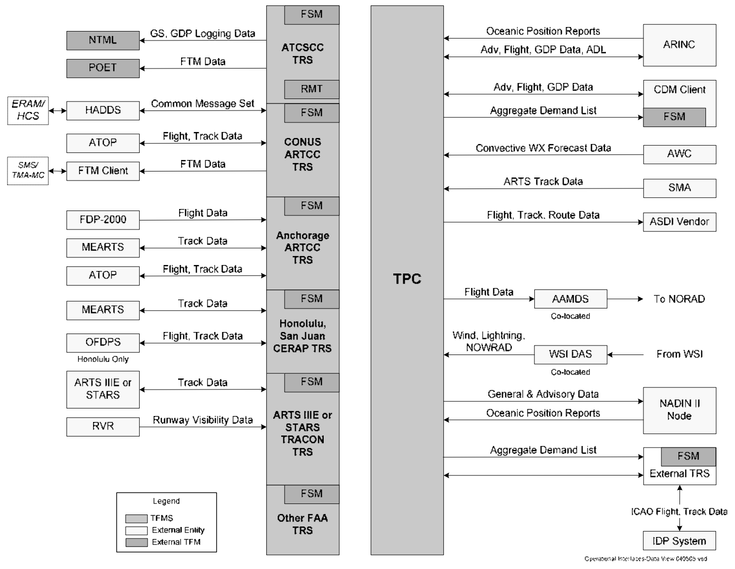
- Cost was medium
  - Specification was less than 10 pages
  - Coding/debugging took roughly three man-months
  - More time spent developing/tailoring scripts and building scenarios
  - Documentation and training were minimal
  - Used in-house Perl programming and air traffic management expertise
- ROI was high
  - Allowed automation of airline inputs to ETMS, which sometimes involved complex integration and reaction to ground delay initiatives
  - Served as a regression test for checking out new builds

TFMS – Traffic Flow Management System

EXPERIENCE. RESULTS.

- **Product under Test:**
  - *Traffic Flow Management (TFM), a major component of FAA Air Traffic Management (ATM), is the strategic planning and management of air traffic demand intended to ensure smooth and efficient traffic flow through FAA controlled airspace.*
  - *TFM seeks to give equitable access for all National Airspace System (NAS) users while accommodating user preferences for flight times, routes, and altitudes to the extent feasible.*
  - *The mission of TFM is to balance air traffic demand with system capacity to ensure the maximum efficient utilization of the NAS.*
  - *A safe, orderly, and efficient flow of traffic, with minimal delays, is fostered through continued analysis, coordination, and dynamic utilization of traffic management initiatives and programs.*

TFMS Operational Interfaces



TFMS

• Make or Buy?

?

TFMS Test Tool - Features

EXPERIENCE. RESULTS.

- **Performs the following tasks** (manual and auto inputs-complete traceability and repeatability)
  - **Simulates all interfaces**
    - Send/Receive messages/data from each interface
    - Inject erroneous messages/data from each interface
    - Be capable of Interacting with the TFM system
      - Respond appropriately to any requests
  - **Adjusts scenario times to system time clock**
  - **Single user interface**
  - **Simulates 1,100 users and 10,000 aircraft**
  - **Drives capacity and performance tests in addition to functional tests**
  - **Reads saved archival data as input, which can be mixed with user inputs**
  - **Simulated flights need to respond and behave in accordance with TFM-issued initiatives** (ground delays, reroutes, ground stops, etc.)
  - **Provides Post-Test Analysis Capability**
- Hardware Status (Original Plan)
  - 10 DL380 servers and 5 workstations identified as TTT test suite
    - 5 servers = HADDS injectors
    - 1 server = DB
    - 1 workstation = post analysis
    - 1 workstation = scenario management
    - 1 workstation = NAS server

    3 servers = HCS, MEARTS, etc injectors
    1 server = Controller
    1 workstation = weather injector
    1 workstation = MQ server

TFMS Test Tool - Single User Interface

EXPERIENCE. RESULTS.

- Graphical interface
- Allow startup/shutdown of the tool
- Control the status of simulated interfaces
- Primarily script-driven
  - Virtually all functions and parameters must be controllable though scripted commands
- Support scenario creation
- Allow single message creation, editing and injection capabilities
- Show test tool activity during operation
- Control all recording/playback capabilities
- Perform test result analysis
- Provide test tool utilities
  - Archival data-to-scenario conversion
  - Scenario manipulation (scenario merge, apply time delta)

TTT – Cost/Benefit

EXPERIENCE. RESULTS.

- Cost was rather high
  - – User Manual was 100+ pages
  - – Coding/debugging took roughly 3 x 24 man-months
  - – More time spent developing/tailoring scripts and building scenarios
  - – Documentation and training are significant and ongoing expenses
  - – Used in-house Java programming and Air Traffic Management expertise
- ROI was high
  - – Basic test scenario served as automated smoke test and is being used by developers, integrators, performance analysts and system testers.
  - – More complex scenarios were required for certain functional tests, e.g. Trajectory Modeling. Tool was enhanced to accommodate this test
  - – Developing new scripts requires minimal skill set

Pros/Cons of Buying

EXPERIENCE. RESULTS.

- **Pro**
  - Complete package available off-the-shelf
  - Training usually available in convenient formats
  - Vendor provides support
  - Tool Usage expertise may be readily available
- **Con**
  - Can be expensive
  - Locked in to vendor’s configuration, control and processing schema
  - Support is almost always necessary
  - As new product requirements are added, may exceed scope of product after making big investment

Pros/Cons of Making It Yourself

- **Pro**
  - Can customize tool to your exact requirements
  - Debugging/diagnosing problems is generally easier/quicker
  - Can expand at will to meet new requirements
- **Con**
  - Can be expensive over long haul (large investment in coding, deployment, maintenance, training and support may be required)
  - May be difficult to recruit new staff to write scenarios for custom product
  - Budgeting may be vulnerable to other project needs
  - Significant software changes may require significant tool modifications

Questions?

Software Quality Group of New England

SQGNE is made possible by the support of our sponsors:

Welcome to SQGNE's 15th 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**
- SQGNE Web site: **www.swqual.com/sqgne/main.html**

Volunteers / Hosts / Mission

Volunteers	Our gracious Hosts
John Pustaver - Founder and Director	Paul Ratty - room, copies, cookies
Steve Rakitin – Programs and web site	Tom Arakel - room, copies, cookies
Gene Freyberger – Annual Survey	Margaret Shinkle - room, copies, cookies
Dawn Wu – our new greeter!!	Jack Guilderson – A/V equipment

SQGNE 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
- ICSQ Nov 9-11 2009 Northbrook, IL www.asq-icsq.org/

SQGNE 2008-09 Schedule

Speaker	Affiliation	Date	Topic
1. Lou Cohen	None	9/10/08	Introduction to using Quality Function Deployment on Software Projects
2. Brian LeSuer	Star Quality	10/8/08	A Survey of Test Automation Projects
3. Howie Dow and Steve Rakitin	None	11/12/08	Estimating using Wideband Delphi Method - An interactive exercise
4. Russ Ohanian	Tizor Systems	12/10/08	Integrating Agile into the Development Process
5. Johanna Rothman	Rothman & Assoc.	1/14/09	Schedule Games
6. Carol Perletz	None	2/11/09	The Nitty Gritty of QA Project Management
7. Robin Goldsmith	GoPro Management	3/11/09	Testing the Untestable
8. Paco Hope	Cigital Networks	4/8/09	Automating security testing of web apps using cURL and Perl
9. Derek Kozikowski	None	5/13/09	Automated Functional Test Design
10. Stan Wrobel	CSC	6/10/09	Test Tool - Make or Buy?
11. Everyone		7/9/09	Annual Hot Topics Night…

Tonight's Speaker…

Test Tool – Make or Buy?
Stan Wrobel, Computer Sciences Corp.

You are faced with the challenge of testing a large software system. One thing you are certain of is that you need to automate much of your testing to really do the job right. There are a multitude of freeware, shareware and commercially sold tools on the market today. Do you elect to buy and/or use one of these or do you choose instead to build your own tool? Stan Wrobel of CSC will share his decades-long experience in testing using both off-the-shelf and custom-built test tools. He will present case studies of his experiences buying, building and using tools for fault-tolerant transaction middleware, consumer websites, and air traffic management systems, then lay out principles for making the decision to make your own test tool or buy.

Bio:

With over 30 years in the computer industry, Stan has served in a variety of roles, including applications specialist, software developer, tester and manager of test teams. Starting out in the Computer-aided Manufacturing industry in 1978, Stan has branched out into fault-tolerant transaction processing middleware, commercial websites and finally into the National Air Traffic Management system. Stan is currently serving as a testing consultant for Computer Sciences Corporation on the Traffic Flow Management Modernization contract for the FAA.