Software Test

Lesson 13
Test Automation
v2.0

Uwe Gühl, Jittat Fakcharoenphol

Fall 2007/ 2008

Thank you

... to Siegbert Voigt, test automation expert, for great support

Contents

- Test Automation
  - Introduction – Heaven & Hell
  - Goal
  - Definitions
  - Philosophy
  - What to automate?
  - Challenges
  - Tools
  - Proceeding

Contents

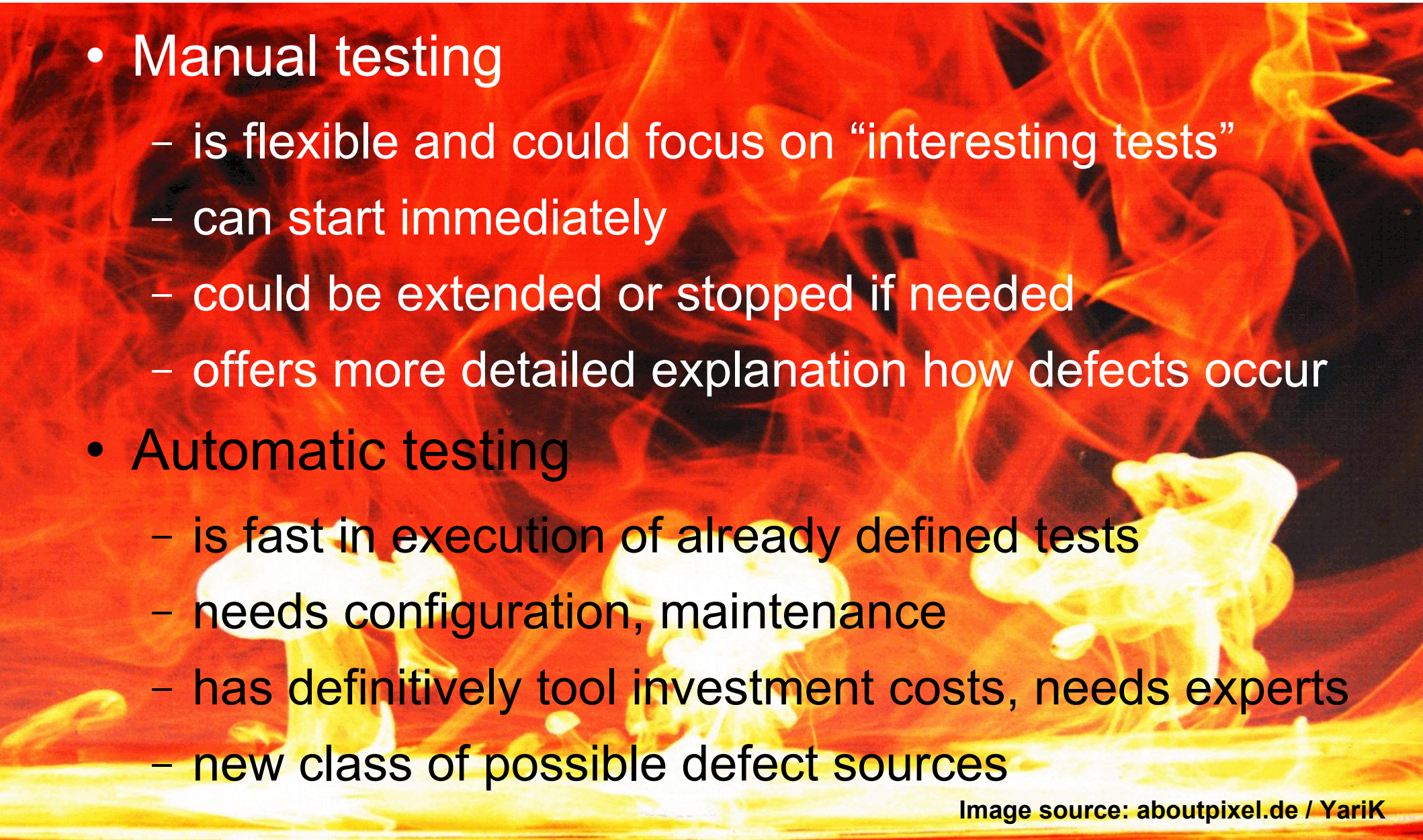
- Test Automation
  - Practical example:
    Testing Web applications with Selenium
    - Web applications
    - Testing web applications
    - Automated testing of web applications
    - Using Selenium
    - Our toy: Testing i5 - a friendship web site
    - Performance testing of web applications
  - Summary

Test Automation Introduction

- Manual testing
  - is slow
  - can be done only a few hours a day
  - expensive, as there are costs of a test team
  - error prone, as humans get tired and do errors
- Automatic testing
  - is typically fast, could be repeated – time saving
  - could be done 24 hours a day, 7 days a week
  - could save money after amortization
  - does not “human errors” - does not get tired

Image source: aboutpixel.de / berliner1017

Test Automation Introduction



- Manual testing
  - is flexible and could focus on “interesting tests”
  - can start immediately
  - could be extended or stopped if needed
  - offers more detailed explanation how defects occur
- Automatic testing
  - is fast in execution of already defined tests
  - needs configuration, maintenance
  - has definitively tool investment costs, needs experts
  - new class of possible defect sources

Image source: aboutpixel.de / YariK

Test Automation Introduction

- Areas of Test Automation
  - White Box Testing
    - Unit tests, build tests
  - Black Box Testing
    - Regression test, smoke test, performance test, stress test, load test, availability test, benchmark test, scalability Test
  - General
    - automated setups, backup, configuration
    - every little tiny tool to help to make testing life easier

Test Automation Introduction

- Capture and replay
  - Often a basic to start
  - Tracing of GUI activities
  - Dependency to stability of the Software under Test (SUT)
  - If you go deeper, you'll find first interesting issues
    - Proprietary scripting languages
    - Access to GUI elements directly or via position
    - Possibility to skip GUI steps

Test Automation Goal

- The goal of test automation should be to reduce the number of tests that need to be run manually, not to eliminate manual testing entirely *(Bret Pettichord)* [Pet01]
- An automatic test tool tries to replace the human drawbacks by the advantages of a robot (*Siegbert Voigt*)

Test Automation Goal

- Goals of Test Automation - high level objectives (*Gerard Meszaros*) [Mes07]
  - Tests should help us improve quality.
  - Tests should help us understand the system under test
  - Tests should reduce (and not introduce) risk.
  - Tests should be easy to run.
  - Tests should be easy to write and maintain.
  - Tests should require minimal maintenance as the system evolves around them.

Value provided by the tests

Characteristics of the tests themselves

Test Automation Goal

- The “Test Automation Manifesto” [MSA03] Automated tests should be:
  - Concise – As simple as possible and no simpler.
  - Self Checking – Test reports its own results; needs no human interpretation.
  - Repeatable – Test can be run many times in a row without human intervention.
  - Robust – Test produces always same result. Tests are not affected by changes in the external environment.
  - Sufficient – Tests verify all the requirements of the software being tested.
  - Necessary – Everything in each test contributes to the specification of desired behavior.

Test Automation Goal

- The “Test Automation Manifesto” [MSA03] Automated tests should be:
  - Clear – Every statement is easy to understand
  - Efficient – Tests run in a reasonable amount of time.
  - Specific – Each test failure points to a specific piece of broken functionality; unit test failures provide “defect triangulation”
  - Independent – Each test can be run by itself or in a suite with an arbitrary set of other tests in any order.
  - Maintainable – Tests should be easy to understand and modify and extend.
  - Traceable – To and from the code it tests and to and from the requirements.

Test Automation Goal

- You must define the goal and the strategy of test automation in your project!

  ... *otherwise you can get lost*

- Decision if and how much test automation must be decided individually for each project
- It is mission critical to use test automation in the right manner

  Ask an experienced test automation developer for advice

  - Should test automation be used at all?
  - When should it be used?
  - Which Test Cases should be automated?

Test Automation Definitions

- Performance Test
  - measures how quickly a system responds under various workloads
  - Given load X, how fast will the system return a result Y?
- Stress Test
  - A test that increases the workload on a system until the system fails
  - Under what load will the system fail and how does it fail?

Test Automation Definitions

- Load Test
  - Determines a system's behavior under various (high) workloads
  - Given a certain load, how will the system behave?
- Availability Test
  - Measures the actual system failure repair time
  - When the system fails, how long will it take for the system to recover automatically?

Test Automation Definitions

- Benchmark Test
  - A simplified, measurable and reproducible test of the basic procedures an application or service will run.
  - May attempt to indicate the overall power of the system by including a mixture of programs or it may attempt to measure more specific aspects of performance (graphics, I/O, computation)
  - Gather task timings and completion metrics for predetermined set of tasks

Test Automation Definitions

- Scalability Test
  - measures the amount of change in linear throughput corresponding to the change in resources
  - If I change characteristic X, how does the performance of the system change concerning a specific criteria Y?
  - Example: “How many more transactions per second can the system handle if I double its memory?”

Test Automation Definitions

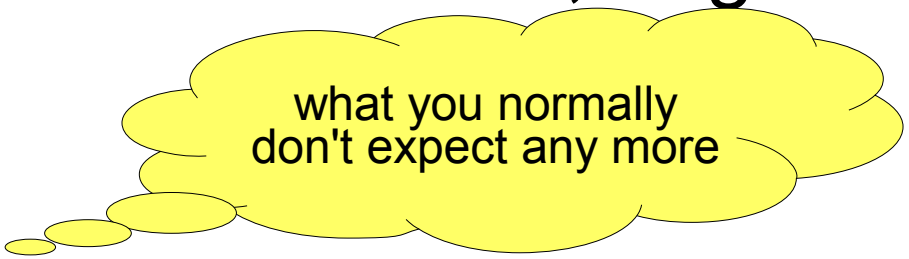
- Reliability Test (Longevity Testing)
  - Under a particular load, how long will the system stay operational?
  - Ensures the product will meet specifications in terms of functionality and length of service
  - Mostly concerned with finding defects and reducing the number of system failures

Test Automation Definitions

- Performance Profiling
  - deep analysis on the behaviour of the system, such as stack traces, function call counts, etc.
  - executes the code in a controlled and specially instrumented environment
  - returns a report listing different statistics like:
    - number of calls to each function
    - time spent in each function
    - heap size over time, etc.
  -

Test Automation Philosophy

- It should be stressed that automation cannot ever be a substitute for manual testing [Kel97]
- Automated tests are very good in data intensive test cases, where only some data combinations result in defects
- With automated tests you can't find new defects in new functions, but because of regression tests “forgotten” defects in old functions, e. g.
  - because of side effects
  - because of wrong builds



Test Automation Philosophy

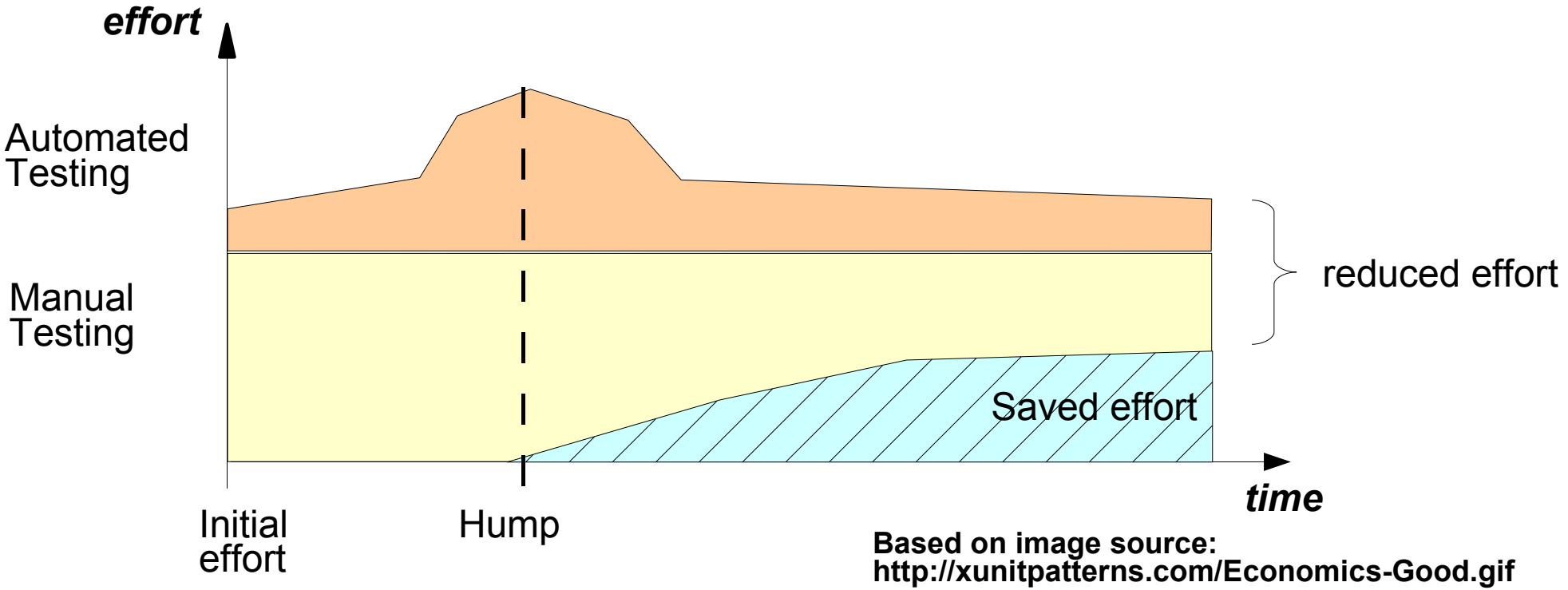
- Controversial statements [p. 93 KBP02]
  - Design your test first, then decide which to automate
  - Design automated tests different from manual tests
- Discussion
  - Automating without good design could result in activities with less value
  - Designing tests with little knowledge of automation will miss chances

Test Automation Philosophy

- Doing test automation
  - First you must first invest in tools and education of the test staff
  - This will result in several new tasks, problems and expenses
- Expectation:
  - The costs over all will decline
  - The quality will rise up

Test Automation Philosophy

- Potential savings in test automation
  - The investment in Test Automation could save in a project a sum ten times higher or more – and a good quality of the software is achieved (S. Voigt)



Based on image source: http://xunitpatterns.com/Economics-Good.gif

Test Automation Philosophy

- Additional value of automated testing
  - Early regression testing establishes a stable software kernel, as it is tested again and again, so that the quality rises

Test Automation
What to automate?

- Candidates for Test Automation [Pet01a]
  - Regression testing
    Very reasonable, if tests have to be run on every build
  - Configuration tests
    Automation may help if many different platforms are supported and all should be tested
  - Setup procedures
    The same setup procedures may be used for lots of different tests. Before automating the tests, automate the setup.
  - Non-GUI testing
    It's almost always easier to automate command line and API tests than it is to automate GUIs.

Test Automation
What to automate?

If you answer yes to any of these questions, then your test should be seriously considered for automation.

- Test Automation Checklist [Kel97]
  - Can the test sequence of actions be defined?
    - Is it useful to repeat the sequence of actions often? E. g. Acceptance tests, Performance tests, Compatibility tests, and regression tests
  - Is it necessary to repeat the sequence of actions many times?
  - Is it possible to automate the sequence of actions? This may determine that automation is not suitable for this sequence of actions
  - Is it possible to "semi-automate" a test? Automating portions of a test can speed up test execution time

Test Automation
What to automate?

If you answer yes to any of these questions, then your test should be seriously considered for automation.

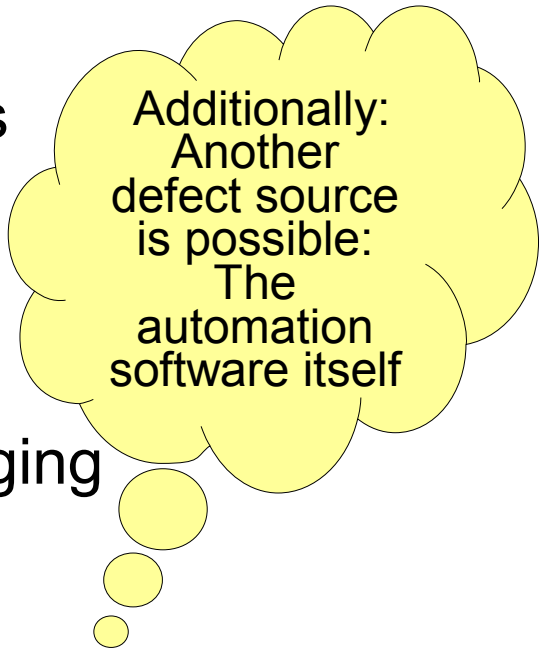
- Test Automation Checklist [Kel97]
  - Is the behavior of the software under test the same with automation as without?
    This is an important concern for performance testing
  - Are you testing non-UI aspects of the program?
    Almost all non-UI functions can and should be automated tests.
  - Do you need to run the same tests on multiple hardware configurations?
    Run ad hoc tests

Test Automation Challenges

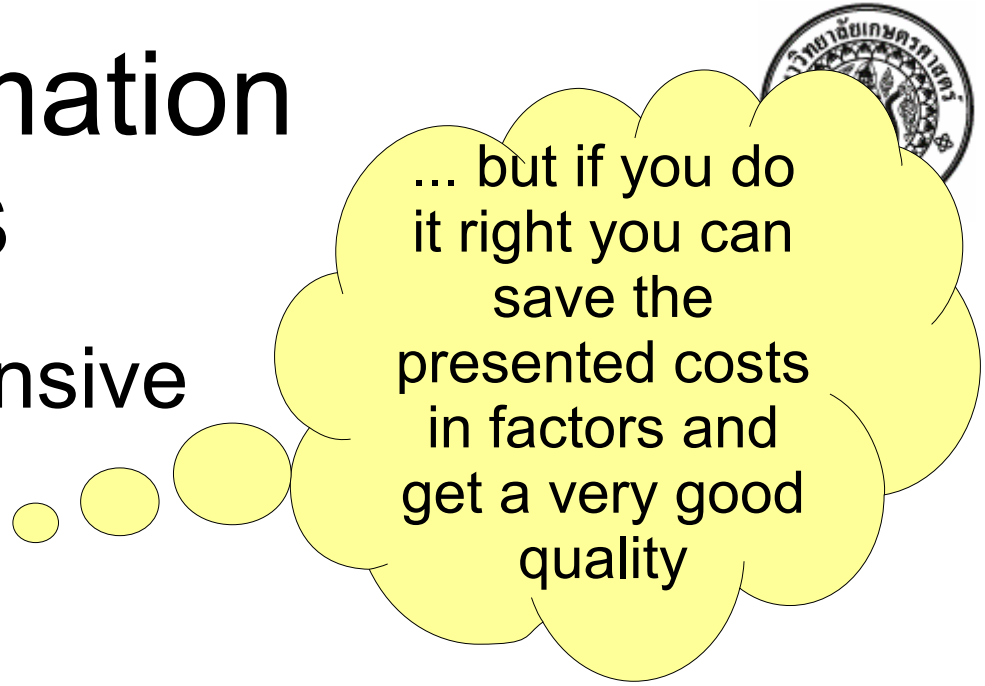
- Reported problems [Pet01a]
  - Lack of clear goals
    Good reasons for doing test automation are to save time, to make testing easier and to improve the testing coverage. But it's not likely to do all these things at the same time
  - Spare time test automation
    Test automation needs time – it must be part of the test strategy and should not be a back burner project
  - Lack of experience
    Junior programmers trying to test their limits often tackle test automation projects – with results often difficult to maintain
  - Technology focus
    How the software can be automated is a technologically interesting problem – but does not help in finding defects

Test Automation Challenges

- To consider in Test Automation [p 196 KFN99]
  - Time Cost
    It takes a long time to create automated tests
  - Testing delay
    Programmers expect fast feedback
  - Inertia
    Challenge to update tests, if the GUI is changing
  - Risk of missed bugs – reported
    - No adequate monitoring of test execution
    - No adequate study of test automation output
  - Partial automation
    Compromise – automate easy tests



Test Automation Tools



- Test automation is expensive
  ... very EXPENSIVE!

- Attention: The main goal of test tool vendors is typically selling test tools.
  Keep this in mind,
  if you get promises concerning
  test automation benefits and simple solutions

Test Automation Tools

- Which Test Automation Tools?
  - For a big project you must calculate 200,000 US $ up to 400.000 US $ for vendor tools
  - So this tools must be selected with care and must be used in the right manner
  - People handling these tools have to have software developer skills
  - Possible alternative: Open Source Tools

Test Automation Tools

- How to decide which tool, which kind of tool to use? Suggestions:
  - Notice requirements
  - Collection of information, play around with tools
  - Intensive evaluation of tools in a realistic environment following defined requirements
    - vendors are normally interested in presenting their tools
    - often it is possible to use a tool a specific time for free
  - Try a proof of concept
  - Recommendation of one tool
  - Decision

Test Automation Tools

- Vendor Tools (selected)
  - Test Automation Tools
    - QTP (Quick Test Professional), Winrunner from HP-Mercury [HPM08]
    - Rational Robot by IBM Rational [Rat08]
    - Silk Test [Bor08]
    - QARun, TestPartner [Com08]

Test Automation Tools

- Vendor Tools (selected)
  - Load Test Tools
    - Loadrunner by HP-Mercury [HPM08]
    - Rational Performance Tester [Rat08]
    - Silk Performer [Bor08]
    - QALoad [Com08]

Test Automation Tools

- Which vendor tool to use? Carey Schwaber, Forrester, May 31, 2006 * wrote:
  "Forrester evaluated leading functional testing solutions — tool suites with support for manual testing, test automation, and test management — across 87 criteria.
  Our research revealed **Mercury Interactive** to be the sole Leader in this market,...
  IBM follows Mercury as a Strong Performer, with especially notable manual testing capabilities and the best test automation tool for users with programming skills.
  Borland Software and **Compuware** are both Strong Performers — but just barely. Our evaluation also included **Empirix**, ..."

* Source: http://www.forrester.com/Research/Document/Excerpt/0,7211,37587,00.html

Test Automation Tools

- Open Source Tools
  - General collection from Mark Aberdour http://www.opensourcetesting.org
  - Goranka Bjedov: Using Open Source tools for performance testing, at Google London Test Automation Conference (LTAC), Sept. 8th, 2006
    - Pdf Presentation: http://www.google.co.uk/intl/en/events/londontesters/speakers.html [Bje06]
    - Google Video: http://video.google.com/videoplay?docid=-6891978643577501895
    - Summary from Alessandro Gentilini with links http://www.flickr.com/photos/agentilini/240878514/

Test Automation Tools

- Open Source Tools (selected)
  - Selenium
    - Automation Tool for Web applications
    - It contains a set of tools that enables you to write test scripts (*in many languages like Java, .net, Perl*) that run through a web browser
    - http://selenium.openqa.org/

Test Automation Tools

- Open Source Tools (selected)
  - WATIR - Web Application Testing in Ruby
    - Automation Tool for Web applications
    - http://wtr.rubyforge.org/

Test Automation Tools

- Open Source Tools (selected)
  - Siege
    - A simple load generating tools for web applications
    - An http regression testing and benchmarking utility
    - It was written on GNU/Linux and does not run under Microsoft Windows
    - http://www.joedog.org/JoeDog/Siege

Test Automation Tools

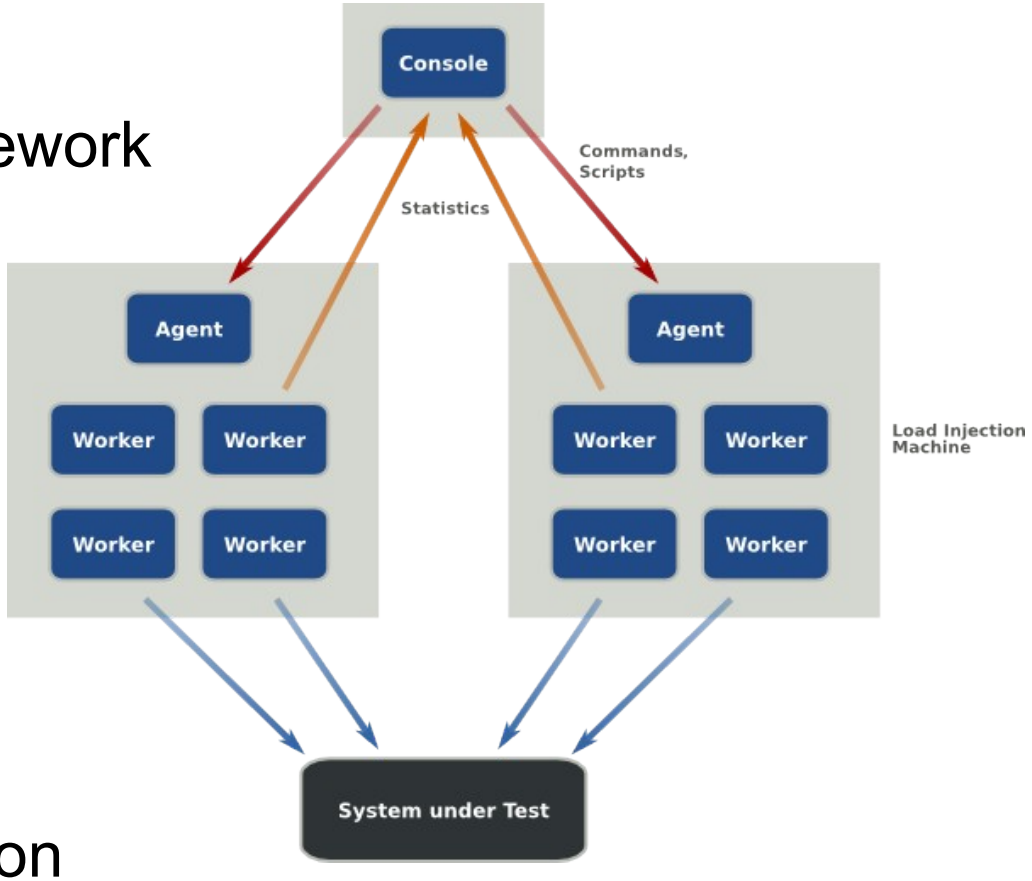
- Open Source Tools (selected)
  - Software Testing Automation Framework (STAF)
    http://staf.sourceforge.net/
  - Eclipse TPTP (Test & Performance Tools Platform Project)
    ... for Java applications.
    http://eclipse.org/tptp
  - OpenSTA (Open, Systems Testing Architecture)
    ... is a distributed software testing architecture designed around CORBA
    http://opensta.org/

Test Automation Tools

- Open Source Tools (selected)
  - Apache JMeter
    is a Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions
    http://jakarta.apache.org/jmeter/

Test Automation Tools

- Open Source Tools (selected)
  - Grinder
    - A Java load-testing framework
    - making it easy to run the activities of a test script in many processes across many machines, using a graphical console application
    - Run load test on various machines
    - Write agent script in Jython
    - http://grinder.sourceforge.net/



Test Automation Tools

- Open Source Tools (selected)
  - Abbot Java GUI Test Framework
    - framework for testing Java GUIs
    - It lets you launch an application or GUI component, play back user actions on it, and examine its state
    - Tests may be coded or scripted where test scripts are JUnit extensions
    - http://abbot.sourceforge.net/

Test Automation Tools

- Vendor Tools
  - professionally developed and maintained
  - could be used immediately
  - usually only small effort concerning the tool itself. Customer could focus on the application to be tested automated
- Open Source Tools
  - are typically tool boxes – effort for customizing
  - normally additional developer effort necessary – need for a team of high qualified people

Test Automation Tools

- Vendor Tools
  - suited for Black Box Testing
  - run typically in an independent application environment
  - results and scripts can be shared and verified
  - No tight dependency to developed software
- Open Source Tools
  - are nearer to software development (e. g. Eclipse TPTP runs in an IDE)
  - aim at White Box / Gray Box Testing
  - are typically language depended (often Java)

Test Automation Tools

- Vendor Tools
  - are very expensive
  - modifications have to be done by vendor (maybe for extra costs) – no access to source code
  - proprietary scripting languages or reduced standard languages (C without pointer)
- Open Source Tools
  - are for free
  - could be modified if needed, as source code is available
  - usually poorly documented
  - support has to be organized – access to one or more experts advisable

Test Automation Tools

- Vendor Tools
  - support many different protocols
  - general support available
- Open Source Tools
  - easy to share results with customers and/or publications
  - scripting in standard programming languages
  - limited support for different protocols
  - steeper learning curve

Test Automation Tools

- Simplified summarization [based on Bre06]
  - You can performance test with open source software, it just takes
    - clarity of purpose
    - infrastructure
    - a production like deployment and
    - time

Test Automation Proceeding

- Simple approach to start performance testing [Bje06]
  - Set up a realistic environment
  - Stress test
    - Check the overload behaviour
    - Find the 80% load point
  - Build a performance test based on the 80%
    - Make it run long enough for a steady state to appear
    - Give it time to warm up at the start
    - Collect the throughput and latency numbers for the Software under Test and the machine performance stats

Test Automation Proceeding

- Activities following the “Performance Testing Guidance for Web Applications” [MFB+07]
  1. Identify Test Environment
  2. Identify Performance Acceptance Criteria
  3. Plan and Design Tests
  4. Configure Test Environment
  5. Implement Test Design
  6. Execute Tests
  7. Analyze, Report, and Retest

Test Automation Proceeding

- Activity 1. Identify the Test Environment.
  - Physical test environment
    - includes hardware, software, and network configurations.
    - a thorough understanding
      - enables more efficient test design and planning
      - helps you identify testing challenges early in the project
  - Production environment
  - Tools and resources available to the test team
  - Process should be revisited periodically throughout the project's life cycle

Test Automation Proceeding

- Activity 2. Identify Performance Acceptance Criteria.
  Goals and constraints concerning
  - Response time – typically a user concern
  - Throughput – typically a business concern
  - Resource utilization – typically a system concern
  - Additionally, identify project success criteria not captured by those goals and constraints; e. g.
    - Performance tests to evaluate what combination of configuration settings will result in the most desirable performance characteristics.

Test Automation Proceeding

- Activity 3. Plan and Design Tests.
  - Identify key scenarios
  - Determine variability among representative users
    - ... and how to simulate that variability
  - Define test data
  - Establish metrics to be collected
  - Consolidate this information into one or more models of system usage to be implemented, executed, and analyzed.

Test Automation Proceeding

- Activity 4. Configure the Test Environment.
  - Prepare the test environment, tools, and resources necessary to execute each strategy as features and components become available for test.
  - Ensure that the test environment is instrumented for resource monitoring as necessary.

Test Automation Proceeding

- Activity 5. Implement the Test Design.
  - Develop the performance tests in accordance with the test design.
- Activity 6. Execute the Test.
  - Run and monitor your tests.
  - Validate the tests, test data, and results collection.
  - Execute validated tests for analysis while monitoring the test and the test environment.

Test Automation Proceeding

- Activity 7. Analyze Results, Report, and Retest.
  - Consolidate and share results data.
  - Analyze the data both individually and as a cross-functional team.
  - Reprioritize the remaining tests and re-execute them as needed.
  - Finish testing of a particular scenario on a particular configuration, if
    - all of the metric values are within accepted limits
    - none of the set thresholds have been violated
    - all of the desired information has been collected

Test Automation Proceeding

- Practical steps for test automation (Siegbert Voigt)
  1. Select the proper Test Cases to automate
     - Avoid frequently changing functionality
     - Avoid functionality not stimulated by mouse or keyboard
  2. Structure the automatic test script(s) to automate all selected Test Cases
  3. Generate the test script(s)
  4. Enter suitable verification points (where you check the responses of the Software under Test)
  5. Optimize the scripts against changes of the Software under Test, mainly in the GUI

Test Automation Proceeding

- Practical hints – How to start in a good way
  - Test Automation is similar to normal software development
  - So act in Test Automation in the same way like in good software development
  - Plan, generate and think in modules, take care on reusability
  - Put often used tests in separate modules, that can be called by other modules

Test Automation Proceeding

- Practical hints – How to start in a good way
  - Do NOT automate straightforward: Select the test cases to be automated carefully
    - Put them small modules: "Think small"
    - ...even smaller.
  - Generate "intelligent" test scripts, where you can skip easily temporary not needed modules

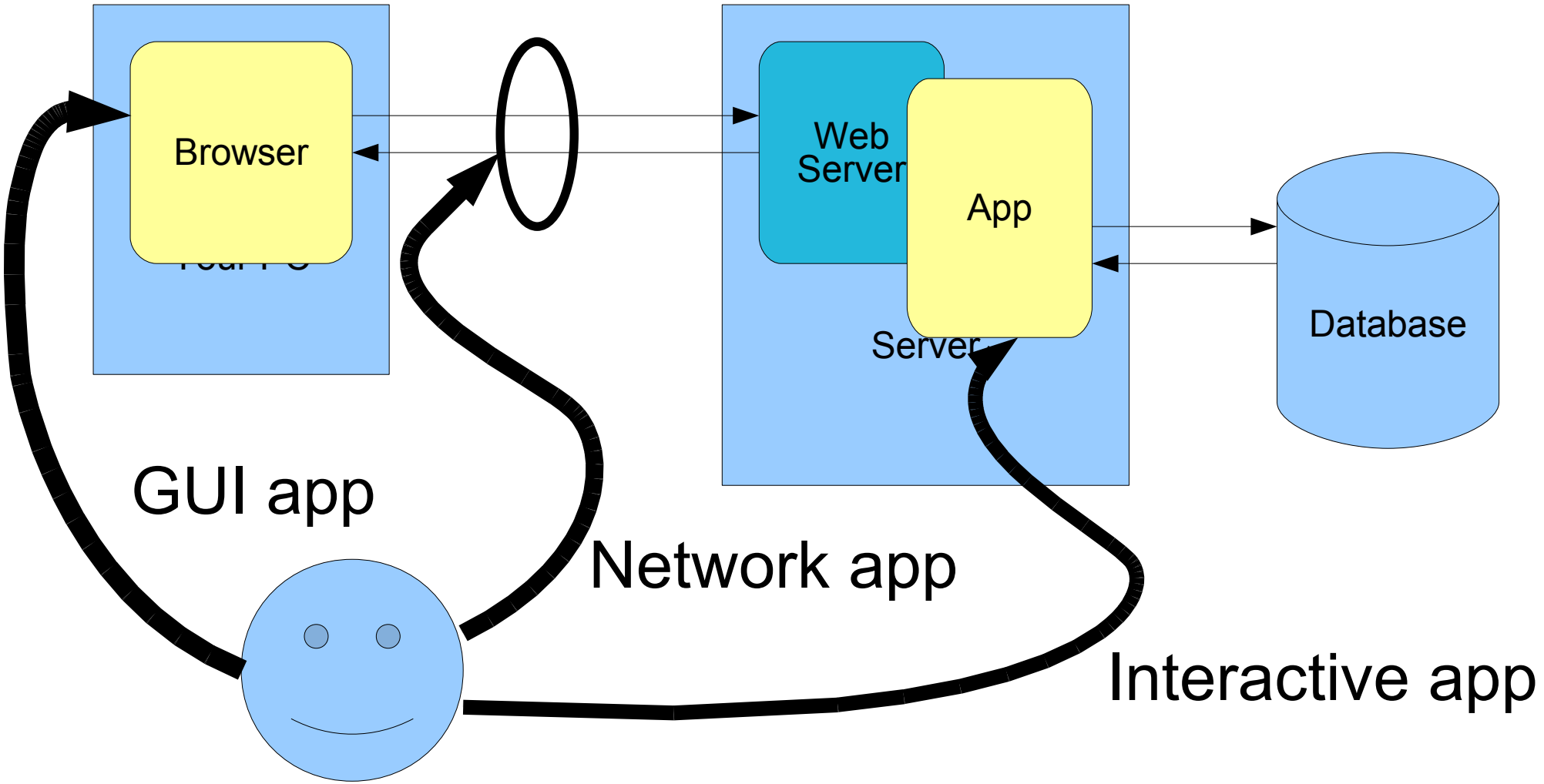
Test Automation Proceeding

- Tradeoff in Test Automation [p. 282 KFN99]
  - Automate as early as possible to recoup the cost of automation
  - Do not automate early, as this time is missed for early manual testing to detect critical defects soon
  - Automate early to higher productivity during peak bug finding period
  - Do not automate early, as the program will change too much
  - Automate early to create a standard acceptance test so you save manually testing time

Test Automation – Practice Web applications

- You put in URL
- You click here, it goes...
- You click again, it moves...
- Interactions with web applications:
  - input actions and page loading
  - a flow of web pages (directed by your input)

Test Automation – Practice
Web applications: what are they?



Test Automation – Practice
Testing web apps in general

- Usually all testing techniques we learn applied
  - You have to think of test cases, test scenarios, etc
- However, the way you interacts with the software changes
  - *In today's practice, we would focus more on this.*

Test Automation – Practice Testing web apps in general

- Functional Testing
  - Can be done at various levels
  - Check the “flow” of your app
- NFR:
  - Performance Testing
  - Load Testing

Test Automation – Practice Automated testing web apps

- You “develop” a test script that drives your web application
  - Again, can be done at various levels
  - The further from the GUI, the easier!
- Develop:
  - Capture and replay
  - Write manually
  - Combination

Test Automation – Practice
Driving your web apps : low level (1)

- Basic scripts (just a list of URLs)

```
http://mywebapp.com/index.html
http://mywebapp.com/login?user=guest&p=hello
http://mywebapp.com/booksearch?q=testing
http://mywebapp.com/addtocart?id=1234
...
```

(shown in red are inputs to the app)

- Sometimes, too rigid for complicated scenarios

Test Automation – Practice Driving your web apps: low level (2)

- How to enter input to web app
  - GET method
    - input in the URLs
    - simple input (shown in previous slide), not secure
    - easy to work with
  - POST method
    - usually used for form processing
    - for complex form, this is very difficult to work with

Test Automation – Practice
Driving your web app: low level (3)

- Various ways to write script, e.g.,
  - list of URLs (no script)
  - HttpUnit, ... (write script in Java, to load pages)

Test Automation – Practice Using Selenium

- Interacts with your app through the web browser
- You can:
  - Write test scripts in various languages
  - Or, use capture and replay technique (with Selenium IDE)
  - Or both (*our practice today*)

Test Automation – Practice Using Selenium

- Getting started
  - The easiest way to start is to use Selenium IDE
  - It runs on Firefox
    - It can be run on other browser as well, but need some work
    - So, if you don't have Firefox, GET IT NOW.
  - Download Selenium IDE at:
    - http://selenium-ide.openqa.org/

Test Automation – Practice
Our toy (1)

- URL: http://158.108.183.10:3000/
- It's a friendship web site: i5
- You can give people comments and fives.
  - What are fives? (I think you know)

Test Automation – Practice
Our toy (2)

- You can:
  - register for an account, log in, edit your profile, and delete your account
  - browse people's profiles, give comments, give fives

Test Automation – Practice Exploration time

- Play around the app
- Notice the URLs and how the pages flow

Test Automation – Practice
First easy test

- Let's start with an easy test: register and login
  - Script:
    - Go to first page: /main/first
    - Click register
    - Put in information, create account
    - Login
    - Click logout
  - Use Selenium to record/replay this

Test Automation – Practice
First easy test

- Assertions
  - How do we know that the test succeeds?
    - Go to first page: /main/first
    - Click register
    - **[put some assertion here]**
    - Put in information, create account
    - **[put some assertion here]**
    - Login
    - **[put some assertion here]**
    - Click logout

Test Automation – Practice Another test

- Profile comment
  - Requirement: “A user can put comment on another user's profile”
  - Script:
    - Log in
    - Pick one user
    - Put in comment
    - Log out

Test Automation – Practice Issues (1)

- how it finds the right links
  - Look at the html code
    - [info]
  - It tells Selenium that the id is 'info-dummy'
  - In general, if this id is not present or the page is more complex you can use XPath to select link, e.g.
    - //[@id=”info-XXXXX”]//a
  - NOTE: Easier to do the test if the app provides you with the information

Test Automation – Practice Issues (2)

- assertions
  - How to check if our comment is there?
  - If we run the test twice and the second run fails, we still notice the previous comment there.

Test Automation – Practice Issues (3)

- Set-up and tear-down
  - We need a way to set up (initialization) and tear down (cleaning up) for each test case
  - One way to do is to provide some testing facility to the application
    - **SHOULD BE REMOVED WHEN THE APP IS DEPLOYED!!!**

Test Automation – Practice Issues (4)

- Set-up and tear-down in our friendship web site i5 with this URLs:
  - /test/create/*username*
  - /test/destroy/*username*
  - /test/users

Test Automation – Practice Performance testing

- Difficult to do with manual testing
  - Need lots of testers
- Need tools that would generate loads
- Demo with Siege
  - NOTE: pay attention on how to POST input

Test Automation Summary

- The topic Test Automation is really complex, already in planning. A lot of experience is necessary
- It's possible to invest a lot of money and to get back nothing

Test Automation Summary

- Rule of thump
  - Big and long lasting projects
    - If you invest time and money the goal of test automation will be reached
    - But first you must invest to get a benefit back.
  - Small or "quick" projects
    - The manual testing is the cheaper, better and quicker way !

Test Automation Summary

- What do I need to get a professional Test Automation Expert?
  1. If somebody asks you:
     "Do you think testing is sexy?"
     ...you should answer : YES
  2. You must have software development skills, because for about 50% test automation is a software development job