Tonight’s Speaker…

Life of a Tester at Microsoft

Urvashi Tyagi

Software Test Manager, Microsoft

You will learn about what a software tester does at Microsoft, how the role interfaces with program managers and developers, what it takes to be successful in this role, various tools we use, and some specifics about Microsoft NERD (New England Research and Development Center)

Bio:

Urvashi currently works at Microsoft as a Senior Test Lead/Manager for Microsoft’s Application Virtualization team based in Cambridge, MA. Prior to joining the team in September 2008, Urvashi was a Team Lead and Test Architect at IBM Software group working on Rational line of products. Prior to IBM, she gained software experience at NuGenesis, a technology startup and at Indian Institute of Management. She is a frequent speaker at sales and user conferences talking about product deployments, testing and debugging. Urvashi holds a M.S degree in Information Systems from Worcester Polytechnic Institute, M.B.A in Finance and Information Systems, and a Bachelor in Engineering in Mechanical and Computer automation systems.

SQGNE Software Quality Group of New England

Agenda

- Software Engineering at Microsoft - Overview
- How we Test software
- Test Tools
- Life of an SDET
- Skills needed to be successful in SDET role
- Life at NERD

Software Engineering at Microsoft - Overview

- Products
- Process
- Organization

Products in Different Categories

260 Products Released in 12 Months

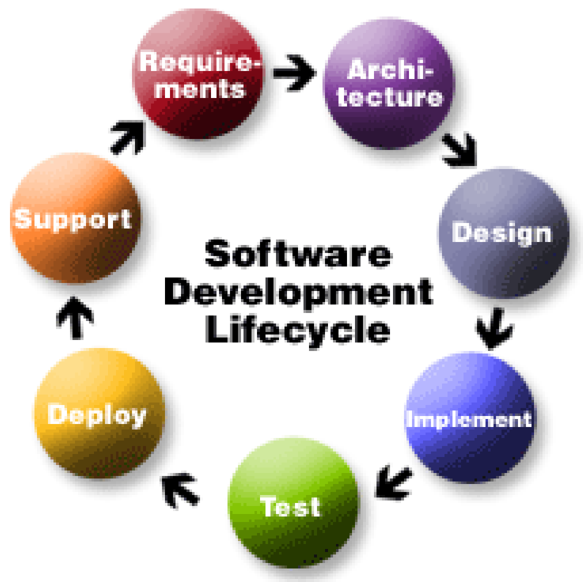
There is no 1 "Microsoft Way"

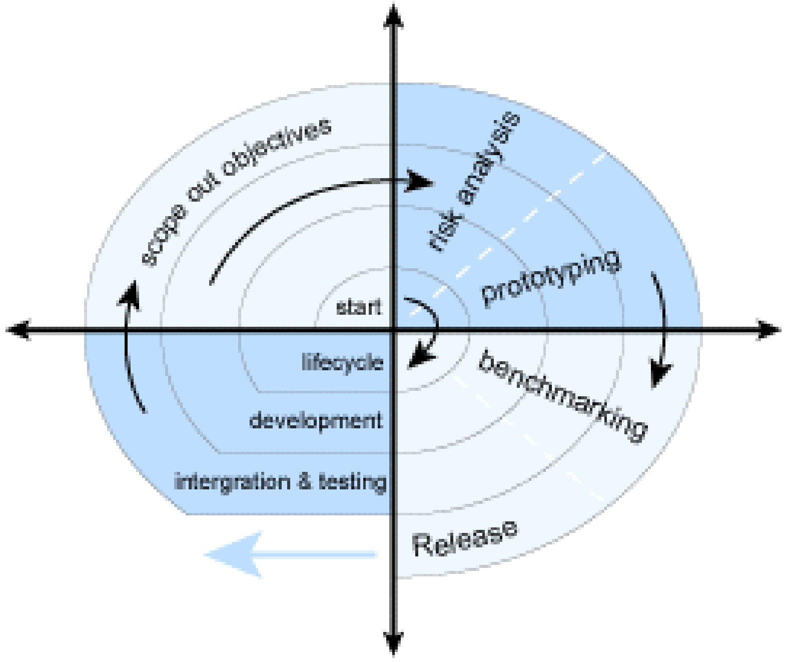
SQGNE Software Quality Group of New England

Software Development Life cycle

Many Methods Used to Develop Products

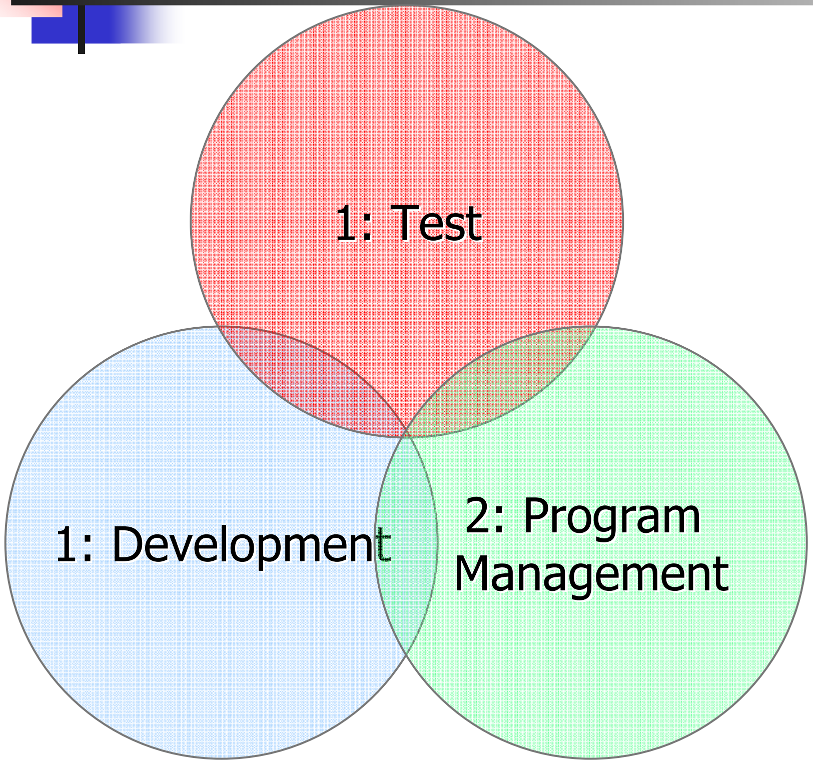
- Waterfall Model
- Iterative Models (Agile, Xtreme, TDD , SCRUM)





Software Quality Group of New England

Organization - the Feature Team Trio



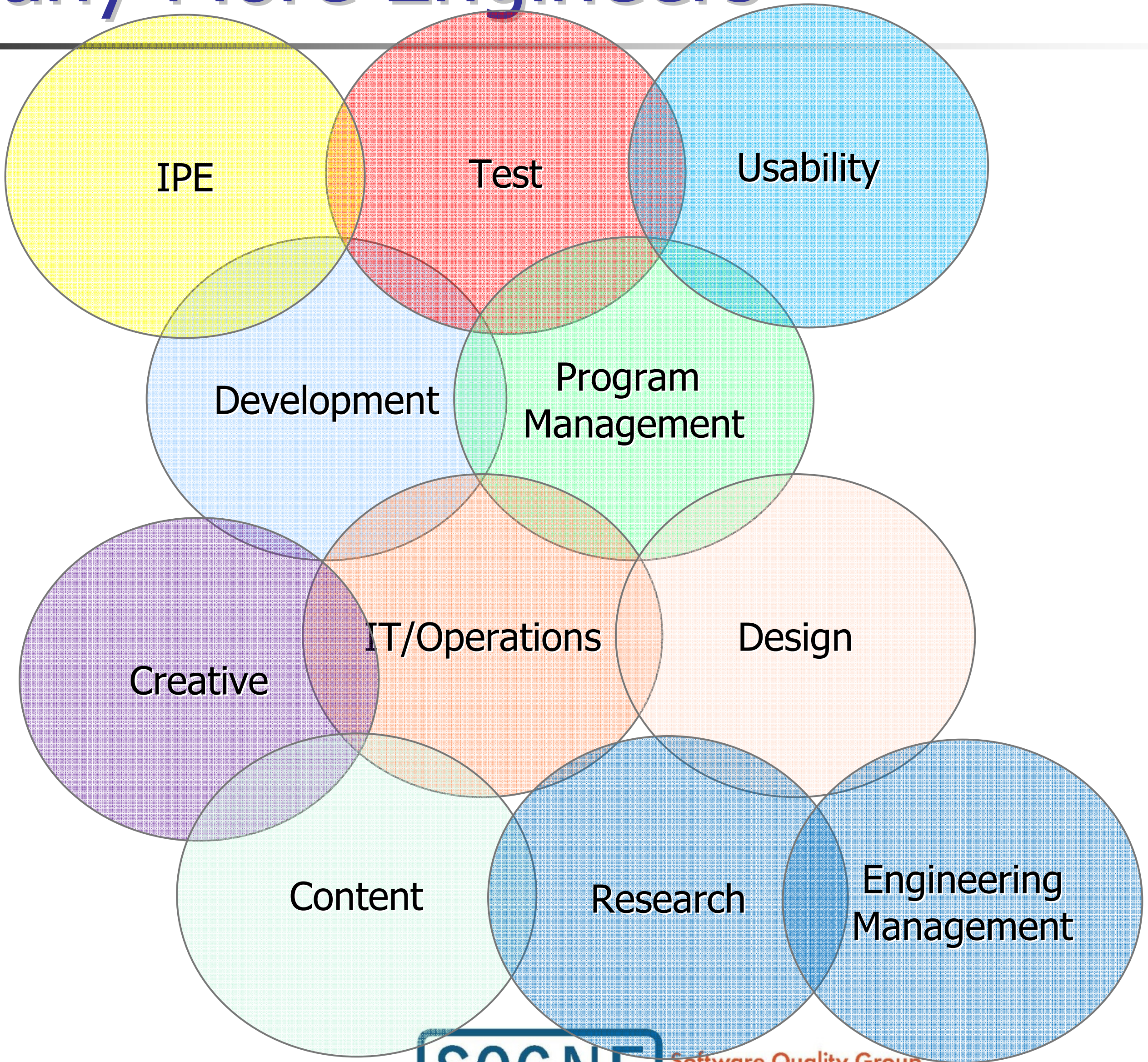
- Program Managers research customer needs, competitors and develop requirements – deliver functional specs
- Developers write product code to the functional requirements – deliver design specs
- Test Engineers (SDETs) ensures the product meets the functional and design requirements – deliver test design spec
- PM-Dev-Test ratio 0.5:1:1

SQGNE Software Quality Group of New England

Dev-Test-PM

Trio Video

Many More Engineers



SQGNE Software Quality Group of New England

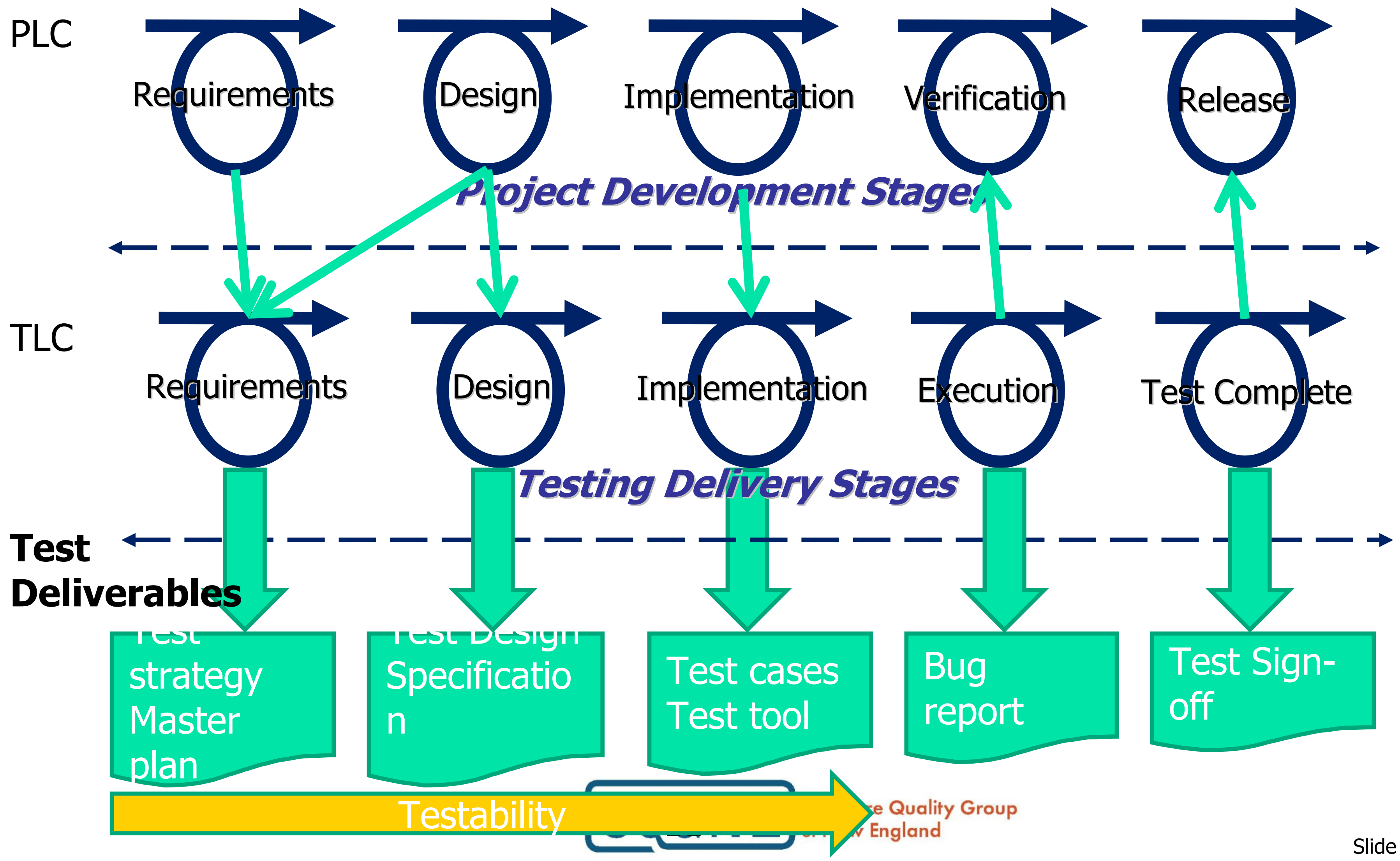
Test team then, and Now

- Tester:
  - STE: Software Test Engineer
  - SDET: Software Development Engineer in Test
- In 2001, 75% STE, 25% SDETS
- In 2002, 100% STE, 0% SDETs
- In 2005, 26% STE, 74% SDETs
- Now: over 95% are SDETs

How we Test software

- Testing life cycle and stages
- Test Techniques taught to new Testers
- Automation focus
- Testability
- Test Design/pattern

Testing Life Cycle and Stages



Test Techniques Taught to New Tester

- Functional testing techniques
  - Boundary value analysis
  - Equivalence class partitioning
  - Combinatorial analysis

- Structural testing techniques and code coverage analysis
  - Decision/Branch coverage
  - Condition coverage
  - Code coverage analysis

- Integration and system level testing
  - Exploratory testing
  - Application compatibility testing
  - Accessibility testing
  - Security testing
  - Globalization testing

- Debugging
- Inspections and Review
- Model Based Testing

Microsoft has Automation Focus

- Windows Vista + Office 12 > 200 MLOC
- Most products target ~70% code coverage before ship
  - Some as high as 90% on unit tests
- Office 12 has over 1 Million automated test cases

- Why Automate so Much?
  - Many products must be supported for 10 years
  - Compatibility regression passes
  - Daily build process increases reuse

Testability

The degree to which components are designed and implemented for test automation to achieve complete code path coverage and simulate all usage situations in a cost-efficient manner.

> Anything that reduces the cost of testing in terms of time and resources is testability

Testability from stage perspective
Planning – Design – Implementation

Dev

- Work with test to define testability features (cyclic)
- Document features and costs in dev design doc
- Review test plan for test approach and coverage

Test

- Work with dev to define testability features (cyclic)
- Document test approach and testability usage in TDS
- Prioritize what you ask for

PM

- Strive for Knowledge such as negative use cases…
- Build schedule that includes testability features
- Review design doc and test plans
- Spec Accessibility features (including MSAA/UIA)

Planning – Design – Implementation

Dev

- Analyze testability requirements, augment them, cost them
- What components can we re-use to reduce cost- do you need that custom control?
- Think about specific design patterns

Test

- Work with dev on testability feature specifics
- Identify possible overlap between testability features and customer scenarios (events, logs)
- Prioritize what you ask for

PM

- Create spec/personas, more detail the better
- Think how will this be tested?
- Identify external dependencies and test concerns/contracts
- Continually feed into the process with customer data

Planning – Design – Implementation

Dev

- Work Items reflect testable chunks
- Work with test to unblock test development
- Fix accessibility bugs

Test

- Incremental test development
- Communicate blocking bugs
- Writing automation earlier using testability features

PM

- Track the schedule
- Provide continual feedback on customer perspective

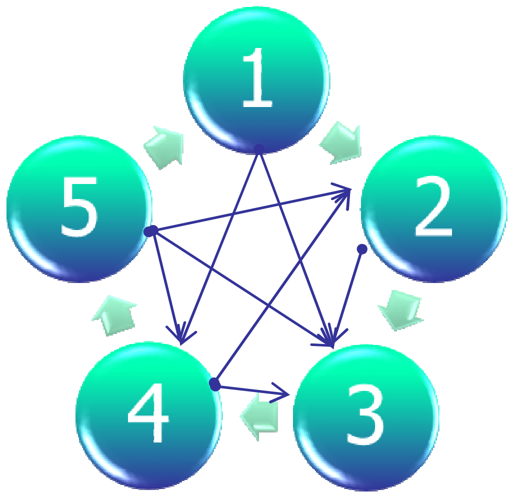
SQGNE Software Quality Group of New England

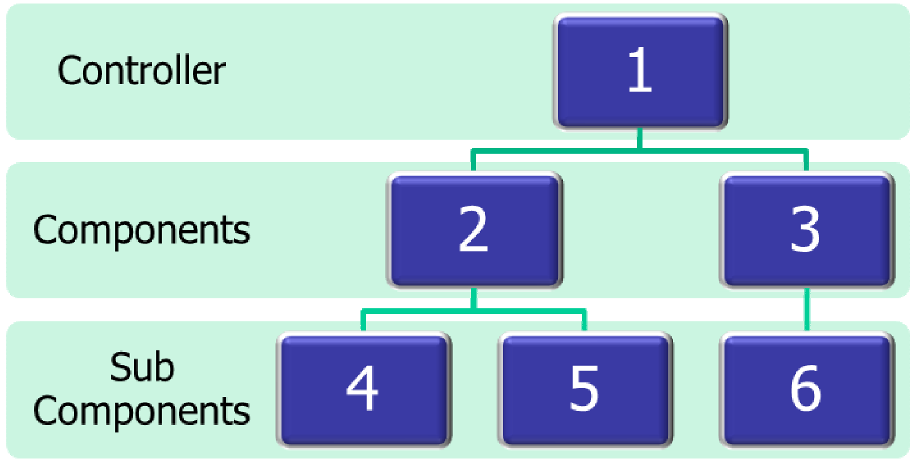
Test Design/Patterns

- Loose coupling
- Mock objects
- Thin UI
- Use SOCK as a guide

Loose coupling

- Good design practice (high cohesion is good)
- Product changes are isolated
- Testing is also isolated to the affected comp and above
- Easier to set up and observe one comp, not whole system





Test goal very clear in each test component

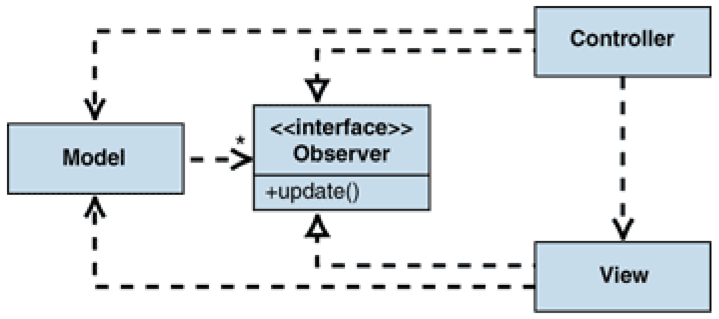
Mock objects

- Replace product subsystems with test subsystems
- Abstract/model internal and external dependencies
- Easier to set up and control the test environment
- Used for unit testing as well
- Can test parts of product without needing all dependent components
- http://www.mockobjects.com

Thin UI

- Separate business logic layer from UI layer
- Historically they have been coupled together
- Allow Test to build logic level and UI level tests
- UI changes do not invalidate the logic level testing
- Easier to drive and observe using a logic layer

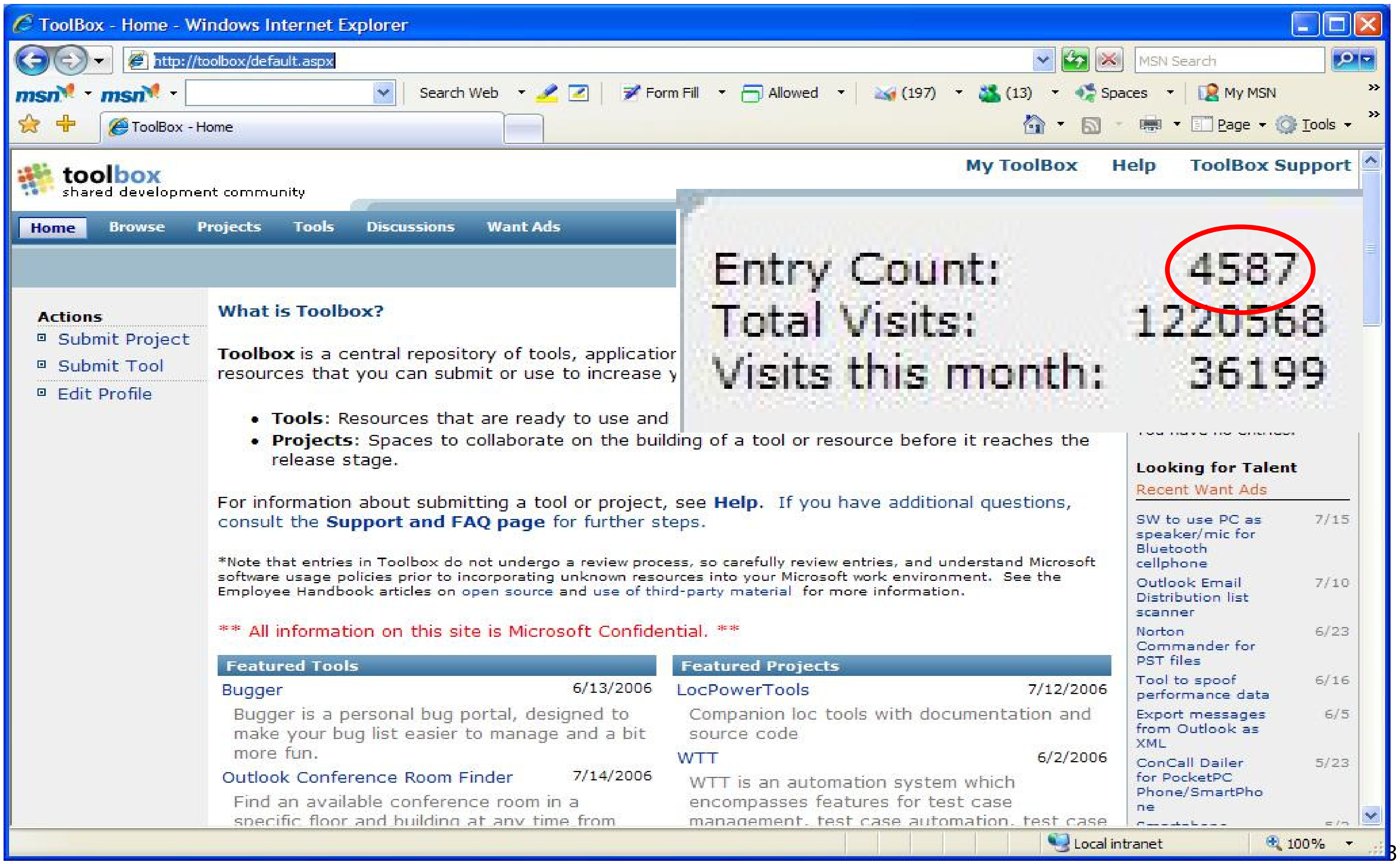
- The Model View Controller design pattern is one example



SOCK Model for Testability

- **Simplicity**
  - Simpler components are less expensive to verify
- **Observability**
  - Ability of automatic tests to watch system or component behavior and reactions
- **Control**
  - Ability of automatic tests to drive the tested system or component through required code paths
- **Knowledge** of expected results
  - Abilities of testers to predict expected results, to know what to observe during the test and to implement the required test logics in automatic tests

Microsoft has a lot of Tools



Entry Count: 4587
Total Visits: 1220568
Visits this month: 36199

Actions

- Submit Project
- Submit Tool
- Edit Profile

What is Toolbox?

Toolbox is a central repository of tools, applicatior
resources that you can submit or use to increase y

- **Tools**: Resources that are ready to use and
- **Projects**: Spaces to collaborate on the building of a tool or resource before it reaches the release stage.

For information about submitting a tool or project, see **Help**. If you have additional questions, consult the **Support and FAQ page** for further steps.

*Note that entries in Toolbox do not undergo a review process, so carefully review entries, and understand Microsoft software usage policies prior to incorporating unknown resources into your Microsoft work environment. See the Employee Handbook articles on open source and use of third-party material for more information.

** All information on this site is Microsoft Confidential. **

Featured Tools

Bugger 6/13/2006
Bugger is a personal bug portal, designed to make your bug list easier to manage and a bit more fun.

Outlook Conference Room Finder 7/14/2006
Find an available conference room in a specific floor and building at any time from

Featured Projects

LocPowerTools 7/12/2006
Companion loc tools with documentation and source code

WTT 6/2/2006
WTT is an automation system which encompasses features for test case management, test case automation, test case

Looking for Talent

Recent Want Ads

SW to use PC as speaker/mic for Bluetooth cellphone	7/15
Outlook Email Distribution list scanner	7/10
Norton Commander for PST files	6/23
Tool to spoof performance data	6/16
Export messages from Outlook as XML	6/5
ConCall Dailer for PocketPC Phone/SmartPhone	5/23

Test Tools

- Source code management
- Test case management (automated and manual)
- Bug management
- Code Coverage
- Performance/Load testing
- Threat Modeling
- Model based testing
- Fuzz testing
- Static code analysis (product code and test code)
- Dynamic analysis

Life of an SDET

Videos

- SDETVIDEOMSCOM_FINAL.mpg

- SDET New Hire

Who is a Tester

- SDET, SDET II, Senior SDET, Principal SDET
- Test Lead II, Senior Test Lead, Principal Test Lead
- Test Manager, Test Director

SDET Tasks

- Write test plans
- Develop test tools, test harness, component mocks
- Develop specialty test tools for security/performance testing
- Automate tests (component, API, GUI)
- Design and document test cases – functional, system
- Execute manual and automated tests
- Find, debug, file, validate bugs
- Participate in design reviews and code reviews
- Provide direct feedback on functional specs, design specs

SDET Skills

- **SDET DNA:** Drive quality upstream, and defect prevention
  - Understand programming concepts, computer architecture
  - Coding skills of entry level developer
  - System and application design (including software architecture)
  - Debugging and root cause analysis
  - Go deep (during component testing) and go broad (during system testing)

SDET Skills..contd

- Engineering competencies
  - Analytical problem solving
  - Customer-focussed innovation
  - Technical excellence
  - Project Management
  - Passion for Quality
  - Strategic insight
  - Confidence
  - Impact and influence
  - Cross-boundary collaboration
  - Interpersonal awareness

SQGNE Software Quality Group of New England

SDET sample Job description

https://careers.microsoft.com/JobDetails.aspx?jid=13908

Basic requirements to be successful in this role include:

Strong recent programming in C++ and C# or Java
Strong history and passion for successful technical problem solving
Solid working knowledge and experience of Windows internals, debugging and software engineering principals
Proven ability to work well with other disciplines and partner teams
Effective communication skills
Proven track record of experience developing software on Windows platforms
3+ years software testing experience with multiple complete product cycles
BS or Masters in Computer Science or comparable experience in the software industry.

Additional experience we like to observe:
Experience working with IT/Enterprise class solutions like System Center Configuration Manager (SCCM), System Center Operations Manager (SCOM), Active Directory, and SQL is strongly desired.

Essential Books for SDET

- How we Test Software at Microsoft
- Testing Applications on the web
- Windows Internals
- Advanced Windows Debugging
- Code Complete
- Writing Secure Code

SQGNE Software Quality Group of New England

About NERD

Microsoft’s New England Research and Development Center, Cambridge

- NERD was founded with the recognition that greater Boston area is home to world's leading universities, cutting edge technology, biotech and healthcare companies as well as a vibrant investment and startup community

- Home to many Microsoft teams:
  - Research
  - Microsoft Application Virtualization (App-V)
  - Future Social Experiences (FUSE) Labs
  - Microsoft Technical Computing

- close to 1,000 employees

Over 250 Events since January 2009

Free meeting space for the tech community

Calendar at: Microsoftcambridge.com

Follow us on Twitter @MSNewEngland

Life at NERD

Online Video - http://microsoftcambridge.com/Default.aspx

Q & A

SQGNE Software Quality Group of New England