


It's Not the Tool's Fault

A Guide to Software Tool Selection for Quality
and Testing Organizations

Lee Barnes, CTO
Utopia Solutions

UTOPIA
SOLUTIONS



**This presentation is based on the following
report from the Software Engineering Institute at
Carnegie Mellon University:**

A Process for COTS Software Product Evaluation

The report is available at:

<http://www.sei.cmu.edu/reports/03tr017.pdf>

Bad Tool or Wrong Tool?



Agenda

- Typical approaches and mistakes
- Evaluation process
 - Plan
 - Define Criteria
 - Collect Data
 - Analyze Results
- Open source considerations

Common Evaluation Methods

- Common methods for evaluating tools
 - Online research
 - Reviewing case studies from similar organizations
 - Obtain subjective information from tradeshows and informal or formal vendor questionnaires and interviews
 - Proof of concept / pilot project

Common Mistakes

- Inadequate stakeholder involvement
- Inadequate criteria definition
- Limited data collection & analysis – relying only on:
 - Internet research
 - Vendor marketing data
 - Non-contextual case studies
 - Past experience and personal preference (emotions)

Common Mistakes - Criteria Definition

Vendor Feature List

Support for 64-bit applications	✓
Extended support for applications using Microsoft Active Accessibility (Sybase PowerBuilder, Microsoft Visual FoxPro and other applications)	✓
UI Automation support (testing Silverlight and other applications)	✓
Testing applications running under another user account	✓
Support for third-party controls (.NET Windows Forms, WPF, MFC, VCL, CLX)	✓
Functional testing (or user interface testing)	✓
Unit testing (including support for MSTest, JUnit, NUnit and DUnit unit tests)	✓
AQtime integration and coverage testing	✓
Keyword testing	✓
Scripting in VBScript, JScript, DelphiScript, C++Script and C#Script	✓
Test debugging (including script debugging)	✓
Event handling	✓
ActiveX objects support	✓
Low-level recording and playback	✓
Data-driven testing	✓
Object-driven testing	✓
Manual testing	✓
Source code control integration	✓

Evaluation Scorecard



Test Tool Evaluation Scorecard

Criteria	TestComplete	QTP	SilkTest
Support for 64-bit applications			
Extended support for applications using Microsoft Active Accessibility (Sybase PowerBuilder, Microsoft Visual FoxPro and other applications)			
UI Automation support (testing Silverlight and other applications)			
Testing applications running under another user account			
Support for third-party controls (.NET Windows Forms, WPF, MFC, VCL, CLX)			
Functional testing (or user interface testing)			
Unit testing (including support for MSTest, JUnit, NUnit and DUnit unit tests)			
AQtime integration and coverage testing			
Keyword testing			
Scripting in VBScript, JScript, DelphiScript, C++Script and C#Script			
Test debugging (including script debugging)			
Event handling			
ActiveX objects support			
Low-level recording and playback			
Data-driven testing			
Object-driven testing			
Manual testing			
Source code control integration			

Extremely biased criteria



Common Mistakes – Data Collection and Measurement

Operating Environment			
Windows XP	x	x	
Windows 7	x	x	
Vista	x	x	
VMWare	x	x	Virtual environments not officially supported for either tool. User forums suggest significant use in virtualized environments (most VMWare)
Application Environment Support			
Standard HTML	x	x	
Terminal Emulator (AS400)	x	x	
Standard Windows	x	x	Custom 3rd party UI objects can be mapped to standard objects in QTP
.Net Winforms	x	x	
Delphi	x	x	Native support for the InfoPower grid not provided by either tool

Difficult to differentiate products



Common Mistakes – Data Collection

Vendor Questionnaire Automated Test Tool Evaluation

Vendor: XXXXXXXXXX

Please provide accurate assessment on the listed criteria

Score each criteria on a scale from 0-5
(0 = no support, 5 = 100% support)

Criteria	Score	Comments
Script Development and Debugging		
Automatically colors code by context	5	
Block indenting and commenting	5	
Execution breakpoints	5	
Step-by-step execution	5	
Step into functions	5	
Step over functions	5	
Add/view watch variables	5	
Modify variables and expressions	5	
Results Reporting		
Step-by-step results reporting	5	
Write custom messages to results	5	
Include screen captures with results	5	

Extremely biased data collection



What's Missing?

- Well defined goals and criteria
- Appropriate techniques for data collection and analysis
- Objective decision making based on relevant data

Tool Evaluation Process

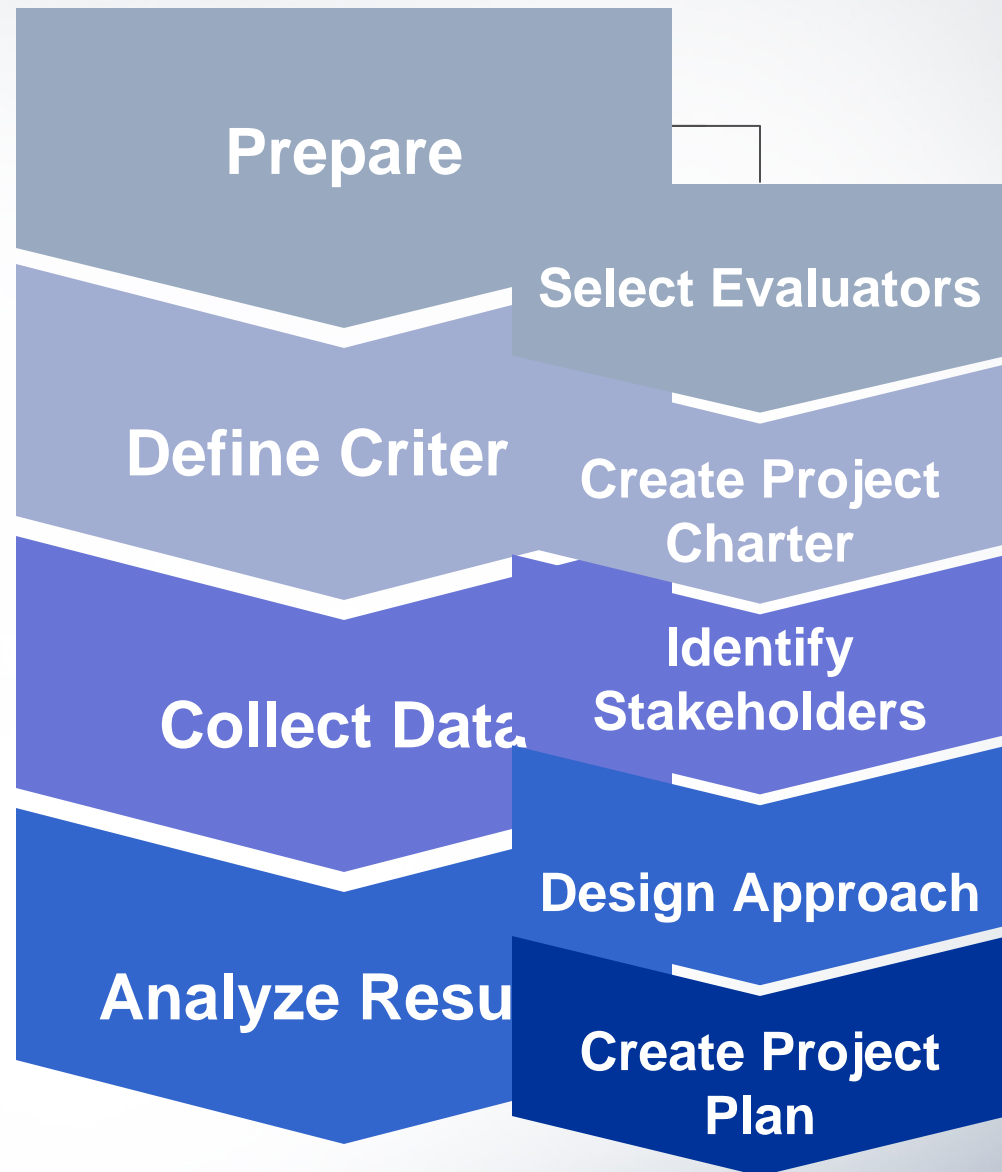
Prepare

Define Criteria

Collect Data

Analyze Results

Evaluation Preparation Activities



Select Evaluators

- Diverse team of skills and perspectives
- Balance of power

Does the technology fit...

\$\$\$\$\$

How strong is the vendor?

We need it now!!!

Document Project Charter

- Simple anchor point for the evaluation

Project Charter - Test Management System Evaluation	
Scope	Top 5 commercial products (measured by market share)
Team	Evaluators: PM, Dev Lead, Test Lead, Developers, Test Analyst Decision Makers: QA Director, App Dev Director
Commitment	Purchase complete 3 months prior to start of Project X
Evaluation Constraints	Budget = \$40-\$60K Operating System = Windows Database = Oracle

Identify Stakeholders

Stakeholder: anyone affected significantly by selection

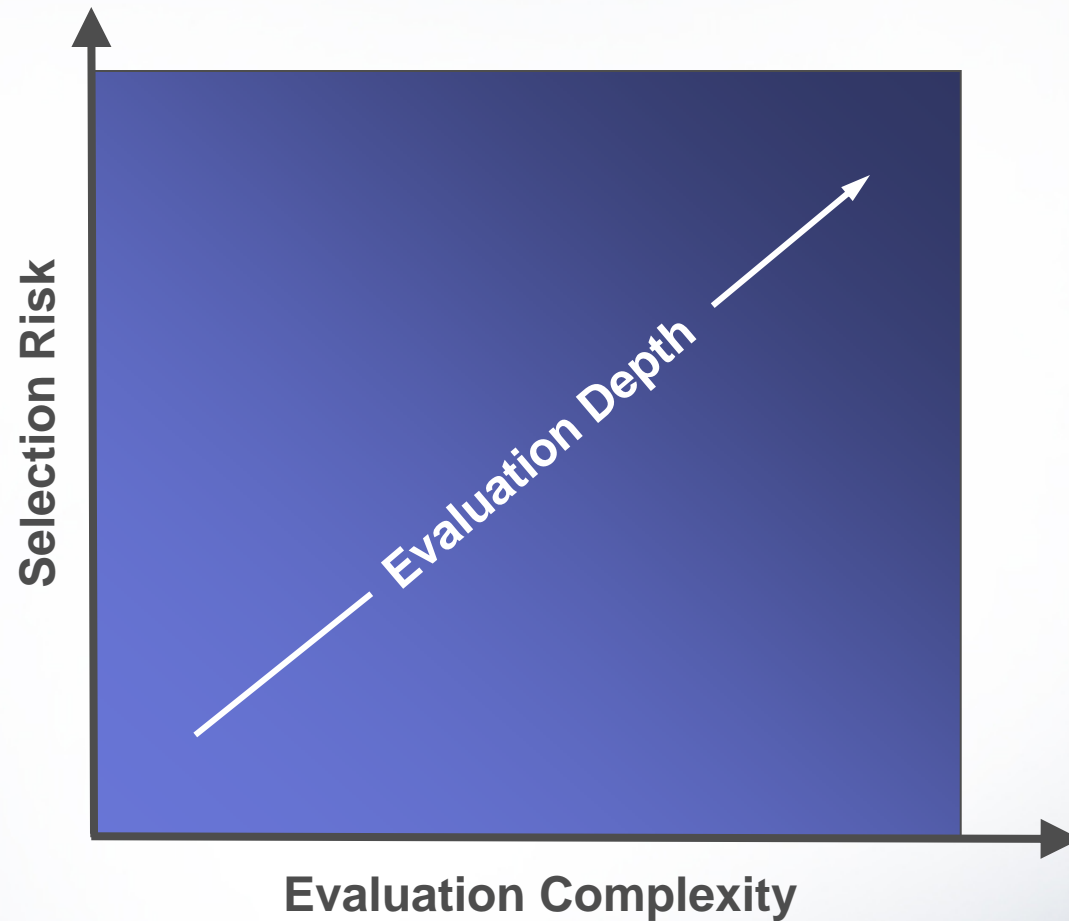
- Key source of requirements
- Better to over-include

Possible Test Management System Stakeholders

- Test Analysts
- Test Leads / Mgrs.
- Developers
- Dev. Leads / Mgrs
- Project Managers
- System Administrators
- IT Operations
- IT Executives
- Business Analysts
- Business Mgrs.

Design Evaluation Approach - Depth

- How deep does the evaluation need to be?

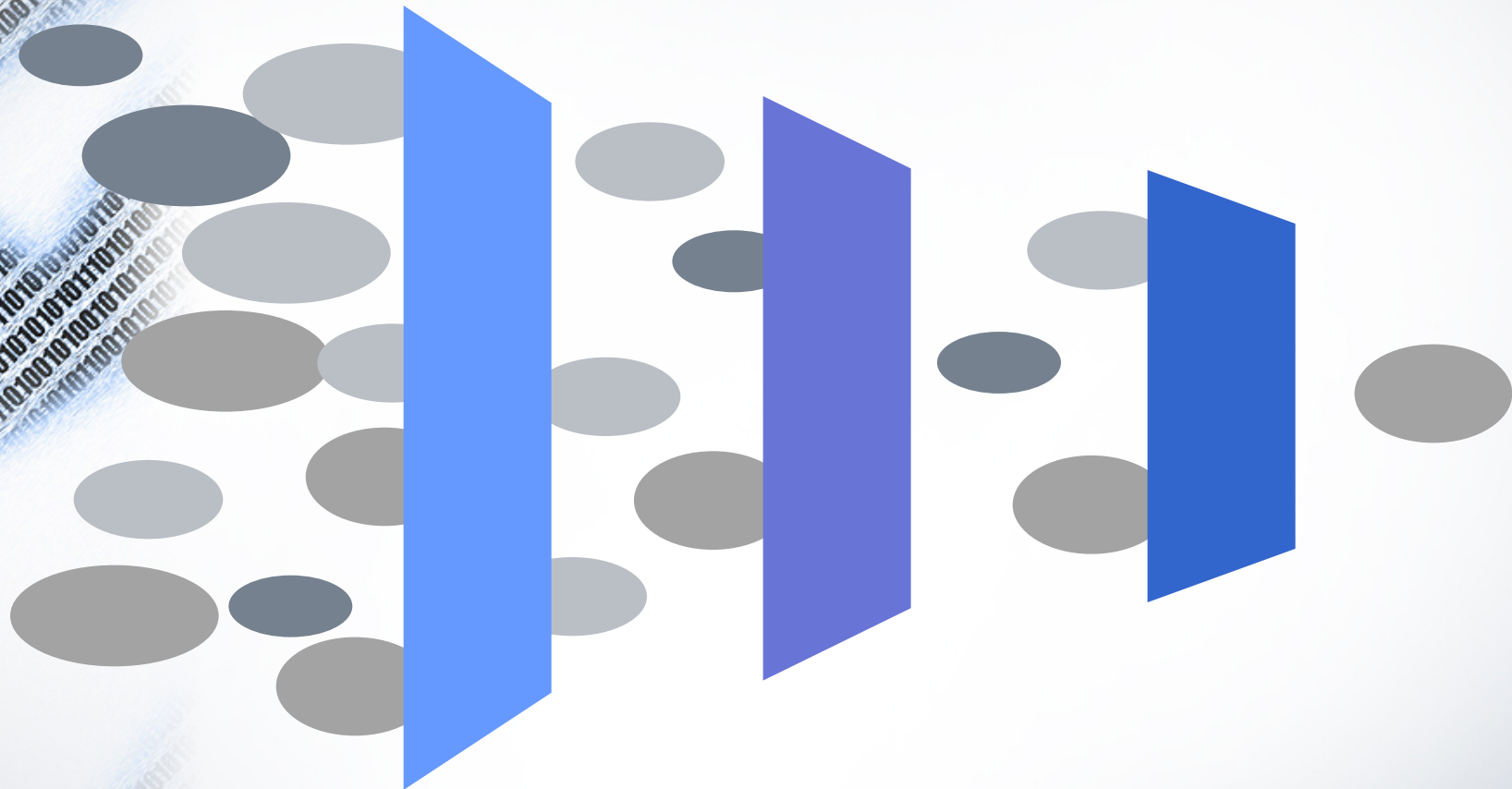


Design Evaluation Approach – First or Best?

- Should you select the first candidate that meets your criteria, or the best?
 - First candidate when little is gained from exceeding minimum criteria
 - *Do not imply that “first fit” means incomplete criteria*

Design Evaluation Approach - Filters

Evaluation Cost 



Cost &
Market Share

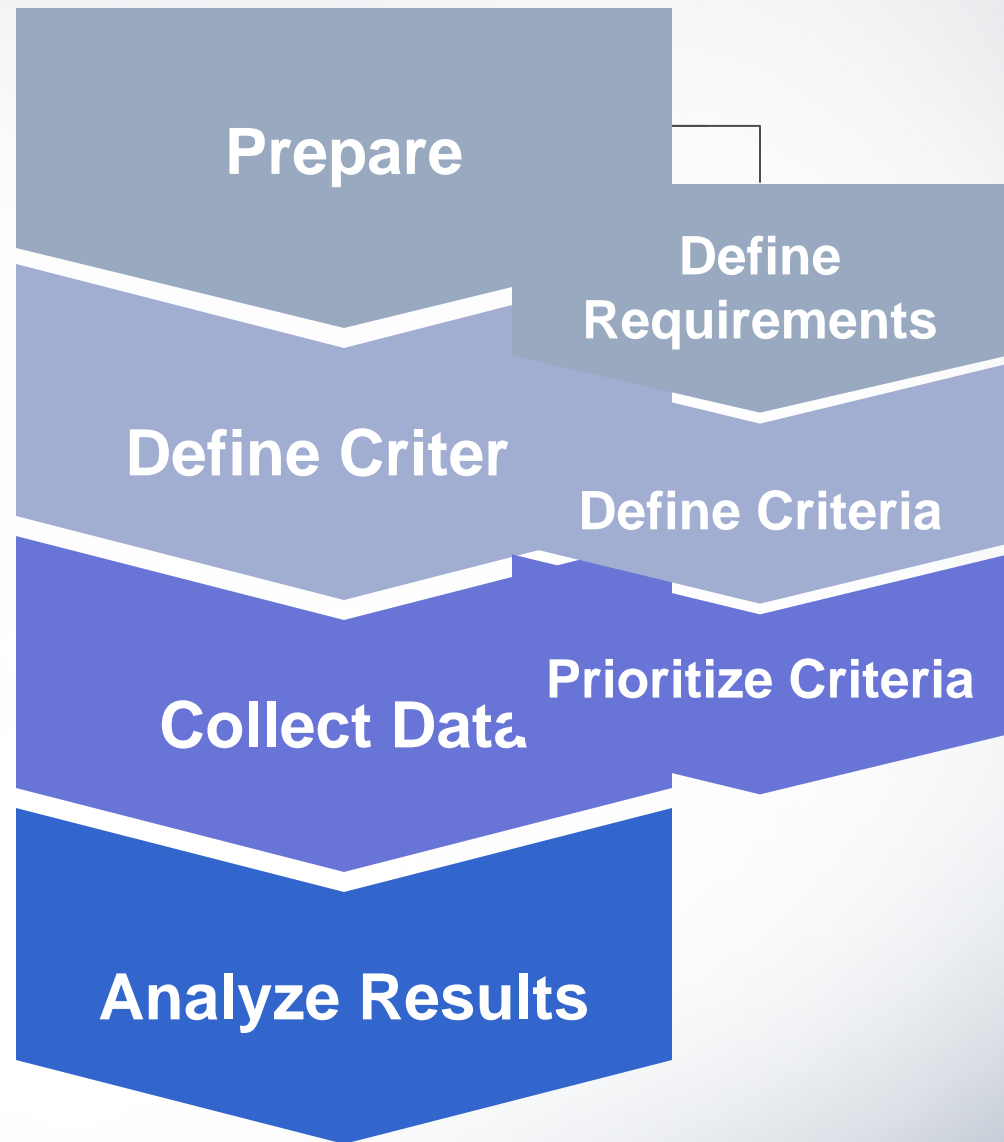
Critical
Criteria

Proof of
Concept

Create Evaluation Project Plan

- Evaluations are often under-planned
- Who? By when?
- Interim milestones
 - **Evaluator selection**
 - **Criteria definition**
 - **Data collection**
 - **Results analysis**
 - **Decision**
- Other planning considerations
 - **Logistics for vendor demos / POCs**
 - **Environment preparation for evaluation software**

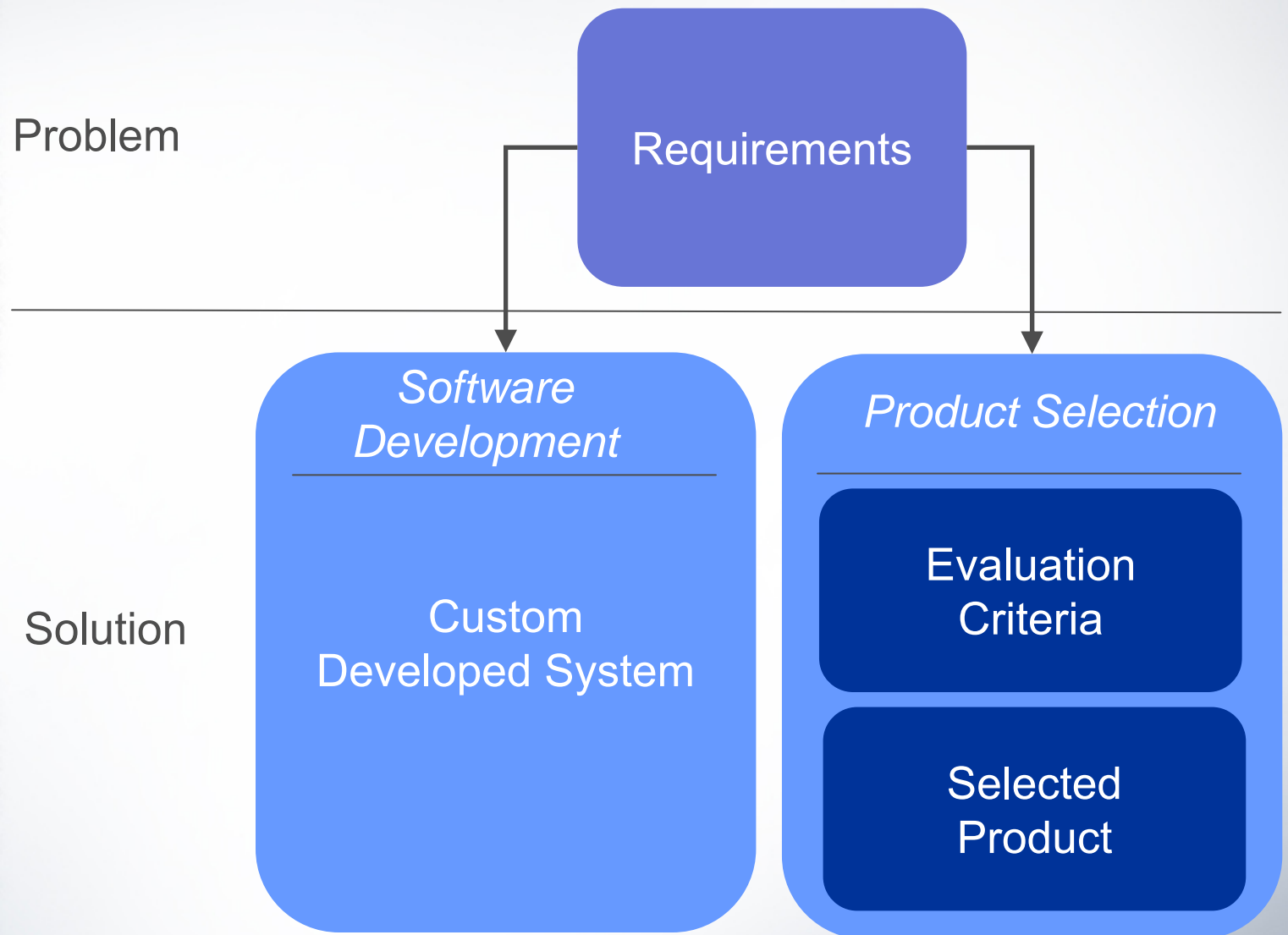
Criteria Definition Activities



Criteria Importance

- Good criteria = good selection
- Don't short-circuit the process by solely using...
 - **Vendor supplied criteria**
 - **Criteria from other organizations / past projects**
 - **Criteria from too few stakeholders**

Requirements vs. Criteria



Requirements to Criteria Example

- Requirement
 - **Test management system shall integrate with enterprise defect management system**
- Criteria
 - **Test management system integrates with ClearQuest in the following manner:**
 - Bi-directional data synchronization
 - Support for all enterprise ClearQuest standard and custom fields

Define Requirements

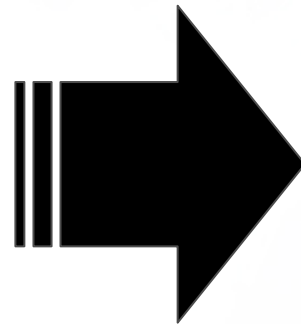
- Traditional Sources
 - Architecture/Integration
 - Organizational/Process
 - Operational Environment
 - Stakeholder Expectations
- Evaluation Specific Sources
 - Evaluation checklists
 - Product feature lists
 - Previous evaluations



Classifying Requirements

- Mandatory – product must meet or be excluded
- Negotiable – failure to meet does not automatically exclude product

100%
Mandatory
Requirements



No
Candidates

Define Criteria

- Good Criteria include:
 - **Clear and measurable description of capability required to meet a need**
 - **Method for measuring a product's ability to meet the criteria**

Criteria and Measurement Example

Criteria

Test management system can be configured to enforce organization's test case review and approval process through the following capabilities:

- Users can be assigned to reviewer and approver groups
- Only users in approver group can approve test cases
- Test cases can be classified in one of the following states: Not Reviewed, Reviewed, Approved
- Only test cases in Approved state can be added to execution cycles

Measurement Method

Vendor assisted POC

Common Criteria Definition Mistakes

- Avoid criteria that have these characteristics:
 - **Difficult / impossible to measure**
 - **Fail to discriminate between candidates**
 - **Irrelevant to the product or context of the evaluation**
 - **Overlap with other criteria**

Criteria Measurement – Method Selection

- Common Methods
 - Document inspection
 - Vendor demos/interviews
 - Hands-on inspection (proof of concept)
- Qualitative or Quantitative?
- Choose method BEFORE data collection begins

Criteria Measurement - Issues

- Measurement error and bias
 - Interpretation of qualitative criteria
 - Evaluation environment issues
- Varying measurement scales

Criteria Measurement – Qualitative or Quantitative Data?

Qualitative Data - Observed

Scale

- 1 / Poor
- 2 / Fair
- 3 / Good
- 4 / Excellent

Definition

- completely unacceptable*
- marginally acceptable*
- mostly acceptable*
- fully acceptable*

Quantitative Data - Measured

Examples

- 2.34 seconds
- 50 GB
- 100 KB/sec
- \$5000

*Quantitative data must
be “qualified” for
analysis*

Criteria Measurement – Data Normalization

- Criteria measured on varying scales is difficult to analyze

Criteria 1		?	Criteria 2	
Allows creation of custom test types	4		Licensing Cost	\$5,000

- Data should be converted to a common scale (normalized)

Criteria Measurement – Data Normalization

Criteria 1 (Qualitative)	
Allows creation of custom test types	4

Criteria 2 (Quantitative)	
Licensing Cost	\$5,750

Normalized Scale

1 = 0.0
2 = 0.3
3 = 0.7
4 = 1.0

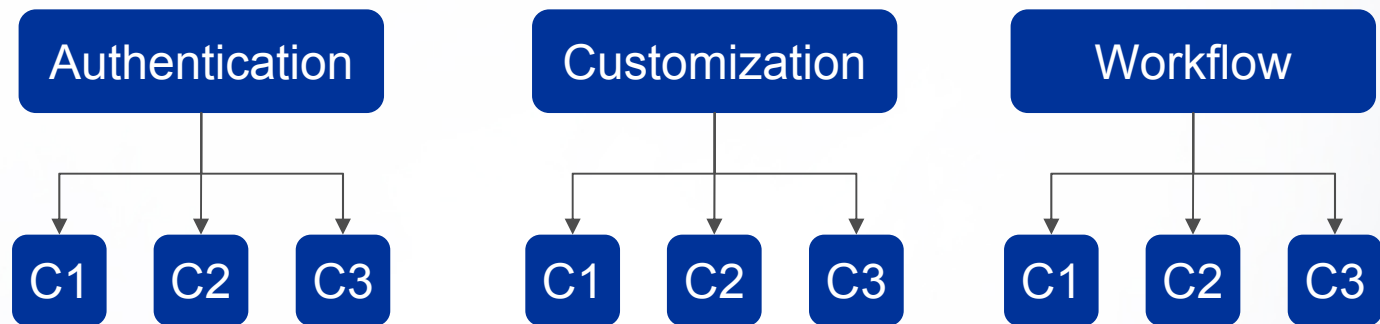
> \$10,000 = 0.0
\$8,500-10,000 = 0.4
\$6,000-8,500 = 0.7
< \$6,000 = 1.0

Criteria 1 (Qualitative)	
Allows creation of custom test types	1.0

Criteria 2 (Quantitative)	
Licensing Cost	1.0

Organizing Criteria

- Organize criteria into similar “parent” groups
- Criteria weighting and data analysis becomes complex with more than two levels



Prioritize Criteria

- Weighting is most common
 - **Unstructured weighting**
 - **Ranking (MAGIQ)**
 - **Pair-wise comparison**
 - **Delphi Technique**
- Unstructured weighting is most common
- Ranking provides a balance of accuracy and efficiency

Criteria Weighting – MAGIQ

- **Multi-Attribute Global Inference of Quality**
 - **Efficient**
 - **Similar results to more formal methods**
 - **Converts rankings to numeric values using the Rank Order Centroid (ROC) concept**
- **Powerful method to compare relative quality**

MAGIQ – How it works

For N criteria, the weight of the Kth criterion is:


$$W_k = \left[\sum_{i=k}^N (1/i) \right] / N$$

Example: Three Criteria

$$W_1 = (1 + 1/2 + 1/3) / 3 = .611$$

$$W_2 = (0 + 1/2 + 1/3) / 3 = .278$$

$$W_3 = (0 + 0 + 1/3) / 3 = .111$$

 For N criteria, there are N weights – the value of these weights is always the same

MAGIQ – Cheat Sheet

2 Criteria

$$W_1 = .750$$

$$W_2 = .250$$

3 Criteria

$$W_1 = .611$$

$$W_2 = .278$$

$$W_3 = .111$$

4 Criteria

$$W_1 = .521$$

$$W_2 = .271$$

$$W_3 = .146$$

$$W_4 = .063$$

5 Criteria

$$W_1 = .457$$

$$W_2 = .257$$

$$W_3 = .157$$

$$W_4 = .090$$

$$W_5 = .040$$

What about tied rankings?

Example: 4 Criteria – C_A , C_B , C_C , C_D

Ranking Results

$$C_A = 1$$

$$C_B = 2$$

$$C_C = 2 \text{ (tie)}$$

$$C_D = 4$$

Average tied values

$$(.271 + .146) / 2 = .208$$

Weights

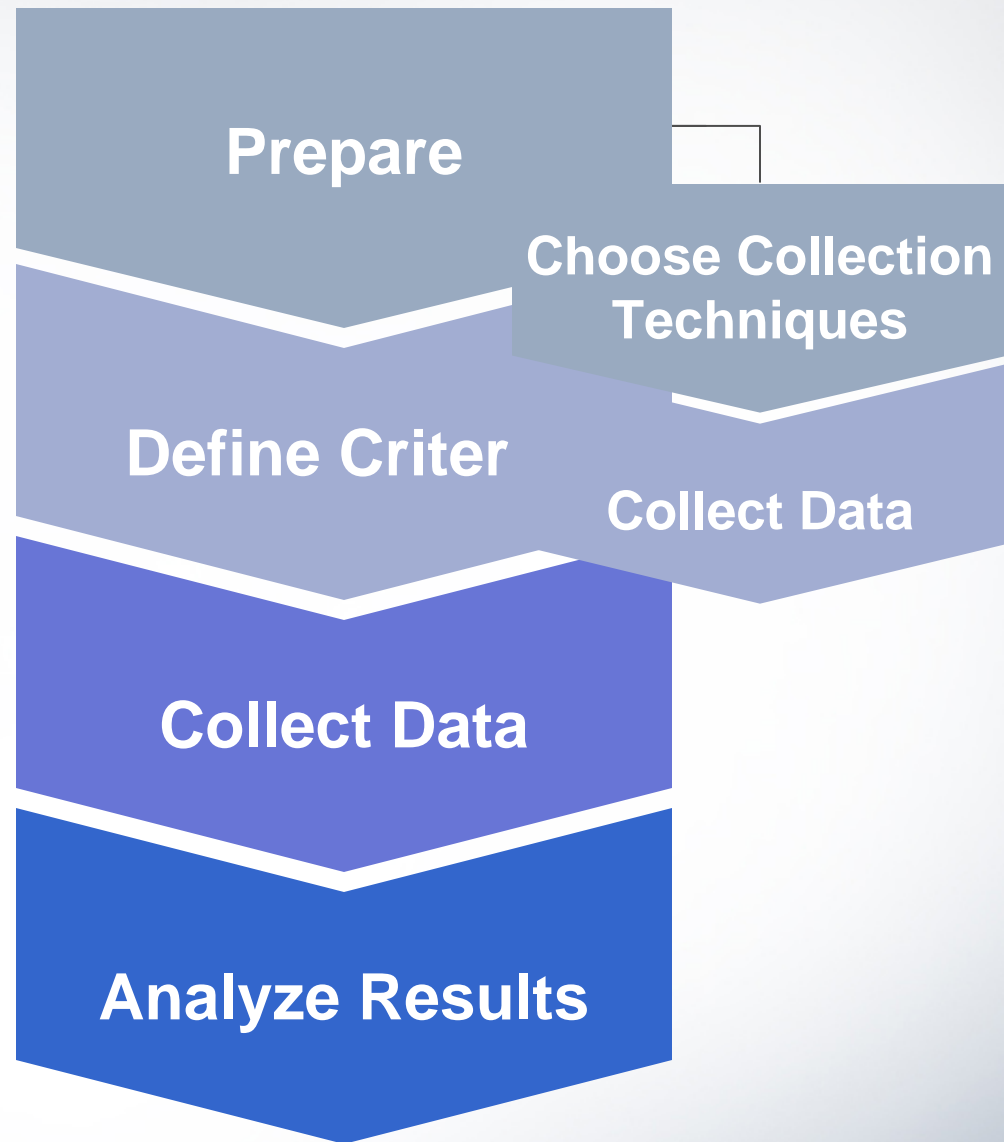
$$C_A = .521$$

$$C_B = .208$$

$$C_C = .208$$

$$C_D = .063$$

Data Collection Activities



Collect Data – Outcomes

Primary

Evaluate products against criteria

Secondary

- Gain understanding of product market place
- Identify new requirements / capabilities
- Learn evaluation expectations are unreasonable
- Identify vendor issues

Collect Data - Techniques

- Documentation review
 - Vendor data sheets
 - User forums
 - 3rd party reviews
 - Product user manuals
- Vendor Demonstration
- Hands-on Inspection
 - Feature-based POC
 - Scenario-based POC

Documentation
Review

Feature POC

Vendor Demo

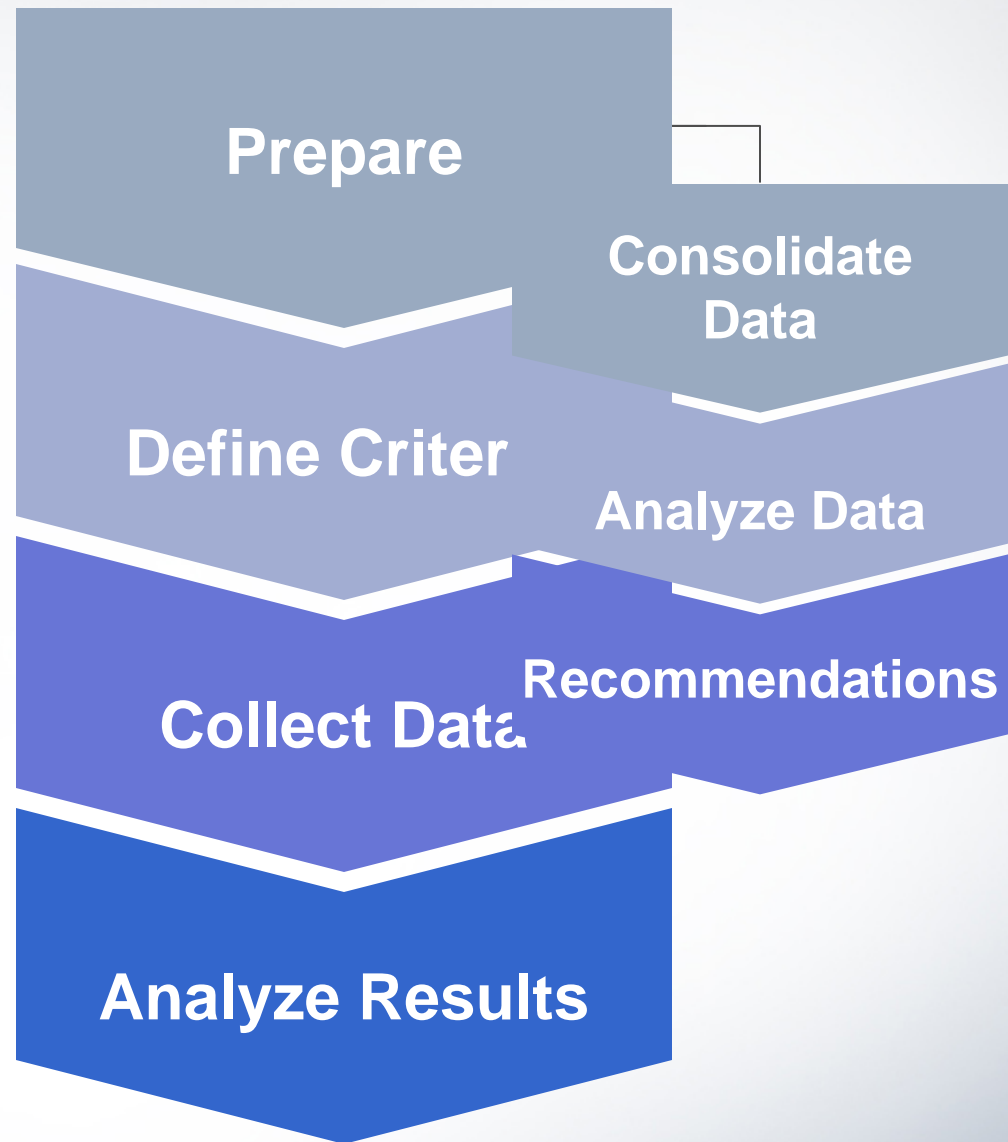
Scenario POC

Data Collection Cost Spectrum

Collect Data – Technique Selection

Criteria Type	Measurement Method(s)
Licensing Cost	<ul style="list-style-type: none">• Vendor discussions• Internet research
Supported operating systems	<ul style="list-style-type: none">• Product documentation
UI object recognition support	<ul style="list-style-type: none">• Product documentation (filter)• Feature-based proof of concept
Configurability to support internal process	<ul style="list-style-type: none">• Product documentation (filter)• Feature-based proof of concept• Scenario-based proof of concept

Results Analysis Activities



Consolidate Data

- Transform raw data into a form that can be more easily analyzed
- Balance between ease of analysis and obscuring data from individual criterion

Don't get caught up in focusing solely on numbers – most evaluation data is based on judgment.

Analysis should include judgment as well



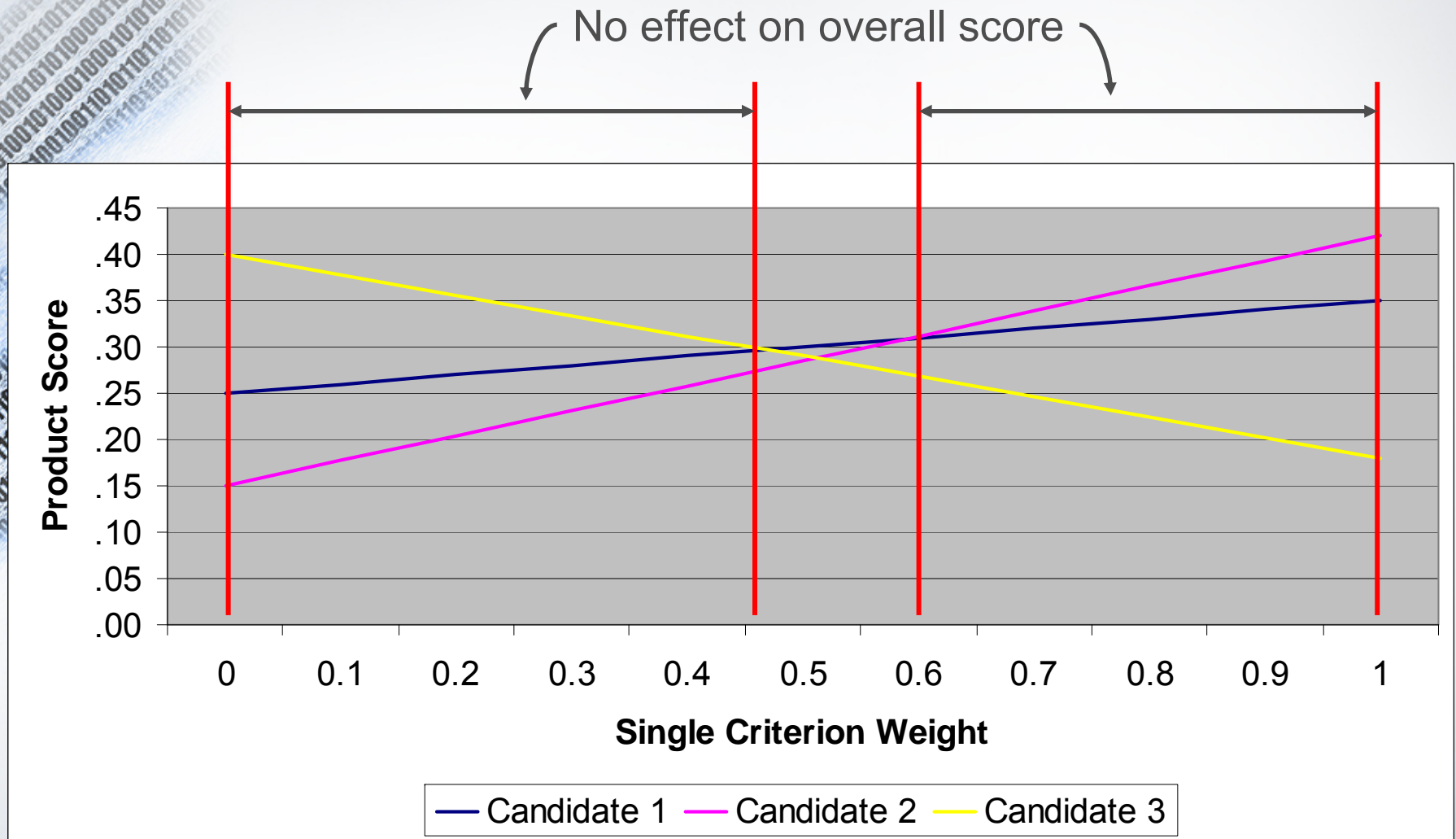
Consolidate Data – Weighted Aggregation

Parent Criteria	Child Criteria	Child		Parent		Final Score
		Score	Weight	Score	Weight	
P1	C1	.1	.15	.610	.35	.561
	C2	.7	.30			
	C3	.7	.55			
P2	C1	1	.32	.780	.26	
	C2	.8	.47			
	C3	.4	.21			
P3	C1	.7	.18	.372	.39	
	C2	.3	.29			
	C3	.3	.53			

Analyze Data

- Evaluations contain uncertainty
 - **Are my weights reasonable?**
 - **Is my collected data accurate?**
- Need techniques to support analysis
 - **Sensitivity analysis**
 - **Gap analysis**

Analyze Data – Data Sensitivity



Analyze Data – Gap Analysis

- Consolidated data can result in similar scores for disparate products
- Need to understand where each product's relative strengths and weaknesses lie
- Gap analysis can be used to focus attention on areas for further analysis.

Analyze Data – Gap Analysis

Criteria	Candidate Scores		
	Candidate 1	Candidate 2	Candidate 3
C1	.8	.5	.8
C2	0	.2	.1
C3	.8	.8	.3
C4	.5	.5	.6
C5	.7	1	.7
C6	.1	.1	.1
C7	.7	1	.2
...			

Data must be normalized



Evaluation Documentation & Recommendation

- Capture high-value information
 - **Industry best practices**
 - **Product capabilities**
 - **Alternative approaches / processes**
 - **Soundness of criteria**
- Possible result is recommend rejection of all candidates

Recommendation – Sample Outline

1. Recommendation Summary
2. Scope of Evaluation
 - a) Evaluation charter
 - b) Products considered
 - c) Evaluation approach (depth, filters, etc.)
3. Evaluation Results
 - a) High-level product features and limitations
 - b) Criteria measurement results for all products
 - c) Confidence in results
4. Evaluation Lessons Learned
 - a) General issues encountered
 - b) Data collection / analysis issues

Open Source Considerations

- Little difference between the evaluation process for open source and commercial tools
 - **Must meet requirements and fit within an organization's technology profile**
- Costs are measured differently
 - **Commercial - upfront licensing fees and ongoing maintenance and support fees**
 - **Open Source – implementation and support through internal staff time or 3rd party vendors**

Open Source Considerations

- Collecting / Measuring support capabilities
 - **Commercial**
 - well defined costs, services & access methods
 - info easily accessible
 - trial calls during POC, customer references
 - **Open Source**
 - Support services / access methods not provided by organization
 - Research community, 3rd party providers
 - Total costs can be difficult to predict

Open Source Considerations

- Filtering open source solutions by “market share”
 - Downloads
 - Adoption rate
 - Forum size and throughput
- Evaluation logistics
 - Generally longer than a commercial tool with similar capability
 - External proof of concept support more difficult to obtain

Summary

- Evaluation is process
- Defining good criteria is the foundation of a successful evaluation
- Avoid common mistakes in criteria definition and data collection
- Use formal data analysis techniques, but don't ignore good judgment
- Recommendation is more than identifying the best product
- Open source and commercial tool evaluations are not significantly different

References

- Botella, P, et al. ISO/IEC 9126 in practice: what do we need to know?. Proceedings of the Software Measurement European Forum (SMEF). Rome. 2004 January.
- Brooks, Lois. Considering Open Source: A Framework for Evaluating Software in the New Economy. ECAR Research Bulletin. 2007 January 2. Vol 2007. No 1.
- Brown, Alan; Wallnau, Kurt. A Framework for Evaluating Software Technology. IEEE Software. 1996 September. Vol 13. No 5. p. 39-49.
- Comella-Dorda, Santiago, et al. A Process for COTS Software Product Evaluation. Pittsburg (PA). Software Engineering Institute, Carnegie Mellon University. 2004 July. Report No.: CMU/SEI-2003-TR-017
- McCaffrey, Dr. James; Koski, Nasa. Competitive Analysis Using MAGIQ. MSDN Magazine [Internet]. 2006 October. Available from <http://msdn.microsoft.com/en-us/magazine/cc300812.aspx>

Questions....

.... and answers

Thank You!

For further discussion:

Lee Barnes, CTO

lee.barnes@utopiasolutions.com