

Review of Programming Languages



Sara Dickinson & Jim Hague / UASG@ICANN60 / 27 October 2017



Universal Acceptance

Reviewing programming languages for compliance with Universal Acceptance good practice

High Level Goals of Study:

- * Evaluate popular programming languages
- * Results  deliver guidance to stakeholders
 - * Library developers (Compliance issues)
 - * App Developers (Best libraries, example code)
 - * Managers (Architectural decisions)
 - * ‘Influencers’ (Input on overall status of UA)
- * How?
 -  * Develop evaluation criteria
 - * Review libraries and
 - * Take remedial action to fix or extend

A Short History of the Work...

Jan 2017	<ul style="list-style-type: none">• First Draft of Evaluation Criteria - high level, inspection based
March 2017	<ul style="list-style-type: none">• After review, updated to use a 'Test Suite' based evaluation.• Initially proposed using UASG004 data (Use Cases for UA)
April 2017	<ul style="list-style-type: none">• First attempts at evaluations of 3 libraries• UASG004 data found insufficient (more on this later)
July 2017	<ul style="list-style-type: none">• Updated with extended test data based on Standards: RFC's and UTS#46 (more work than expected).
August 2017	<ul style="list-style-type: none">• Version 1.0 released.• Feedback still needed based on full evaluations.

Overview of Evaluation Criteria

1

Basic Information

Description of library, test environment, etc. *(Not Scored)*

2

Auxiliary Information

Support + maintenance, bug tracking, maturity, portability

3

Implementation notes

How easy it is to use in practice, any issues with the documentation

4

Test Suite

- **IDNA2008 (Unicode \longleftrightarrow ASCII, Registry conversion, DN equivalence)**
- **Syntactic checks and decomposition for DN, URL, IEA**
- Semantic checks for DN, URL, IEA (Not scored)

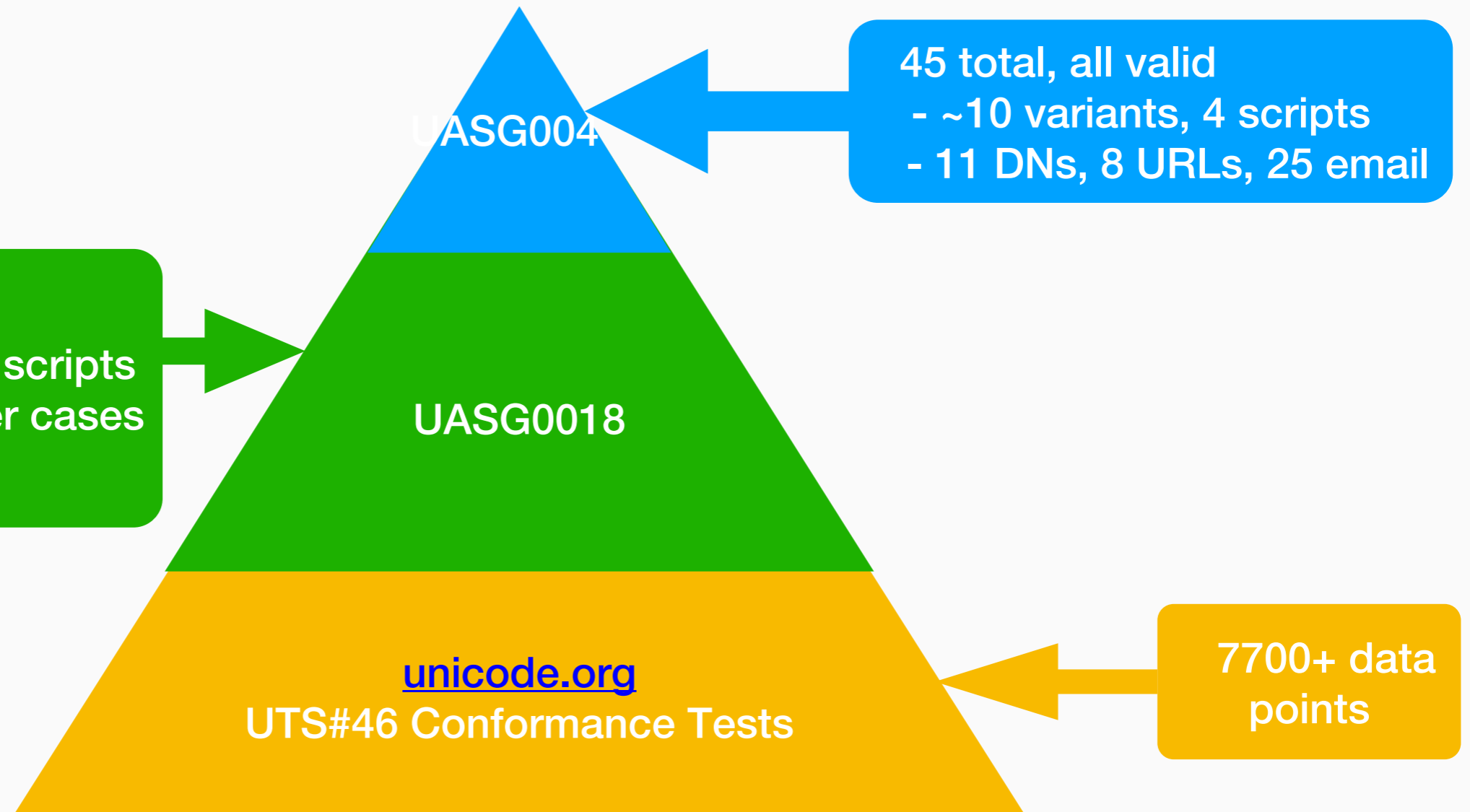
5

Mitigation Actions

Missing functionality, bug reports, documentation issues

Test Suite Data

- A broad spectrum



Hope the UASG0018 is a good middle ground

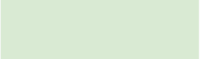
Evaluations done to date

What is a library?

- * A 'set of libraries/frameworks' that provide UA compliant functionality in a given language
- * First evaluation - chose 3 languages:

Javascript	Java	C		
<code>punycode + isemail + uri-js</code>	<code>Apache Commons</code>	<code>libidn</code>	<code>libidn2</code>	<code>libicu</code>

Evaluations done to date

 Functionality available

Language	JavaScript	Java	C		
Library/Libraries	punycode + isemail + uri-js	Apache Commons	libidn	libidn2	libicu
Unicode to ASCII (IDNA2008 lookup)	4	4	4	5	5
Unicode to ASCII (IDNA2008 register)				5	5
ASCII to Unicode	4	4	4	5	5
IDNA2008 domain name comparison					2.5
Domain name syntactic check		3*			
Domain name decomposition					
Email address syntactic check	2**	TBC			
Email address decomposition		TBC			
URL syntactic check	3	TBC			
URL decomposition	5	TBC			
Test suite total (D)	18	11	8	15	17.5

Evaluations done to date

- * *Only partially completed - used UASG004 data (not published)*
- * Only Test Suite results shown (scores out of 5)
 - * Coverage varies...
 - * Data not granular enough
 - * Issues found: static TLD list*, email not recognise U-label domains**

What next?

- * Need to **perform full evaluations** using latest criteria of several languages
 - * There is still uncertainty on the coverage and scope of remedial action
 - * Who? Evaluation requires *software testing skills* as a minimum
 - * Contribute test suite back to library?
- * **Who performs the remedial action** - library maintainer or evaluator?
 - * Would require good will from library maintainer, but may be motivated by UASG ‘stamp of approval’
 - * If evaluator, would require *software development skills*
 - * Bug fixes can be one line fix or significant re-write
 - * Implementation of missing features?
- * **Bug bounty model?** Doesn’t directly recognise ‘good’ libraries.



Links

- * [Programming-Languages-Evaluation-Criteria.pdf](#)
- * [UASG004](#)

