Combinatorial Testing on Implementations of HTML5 Support

Xi Deng, Tianyong Wu, Jun Yan and Jian Zhang

Institute of Software, Chinese Academy of Sciences
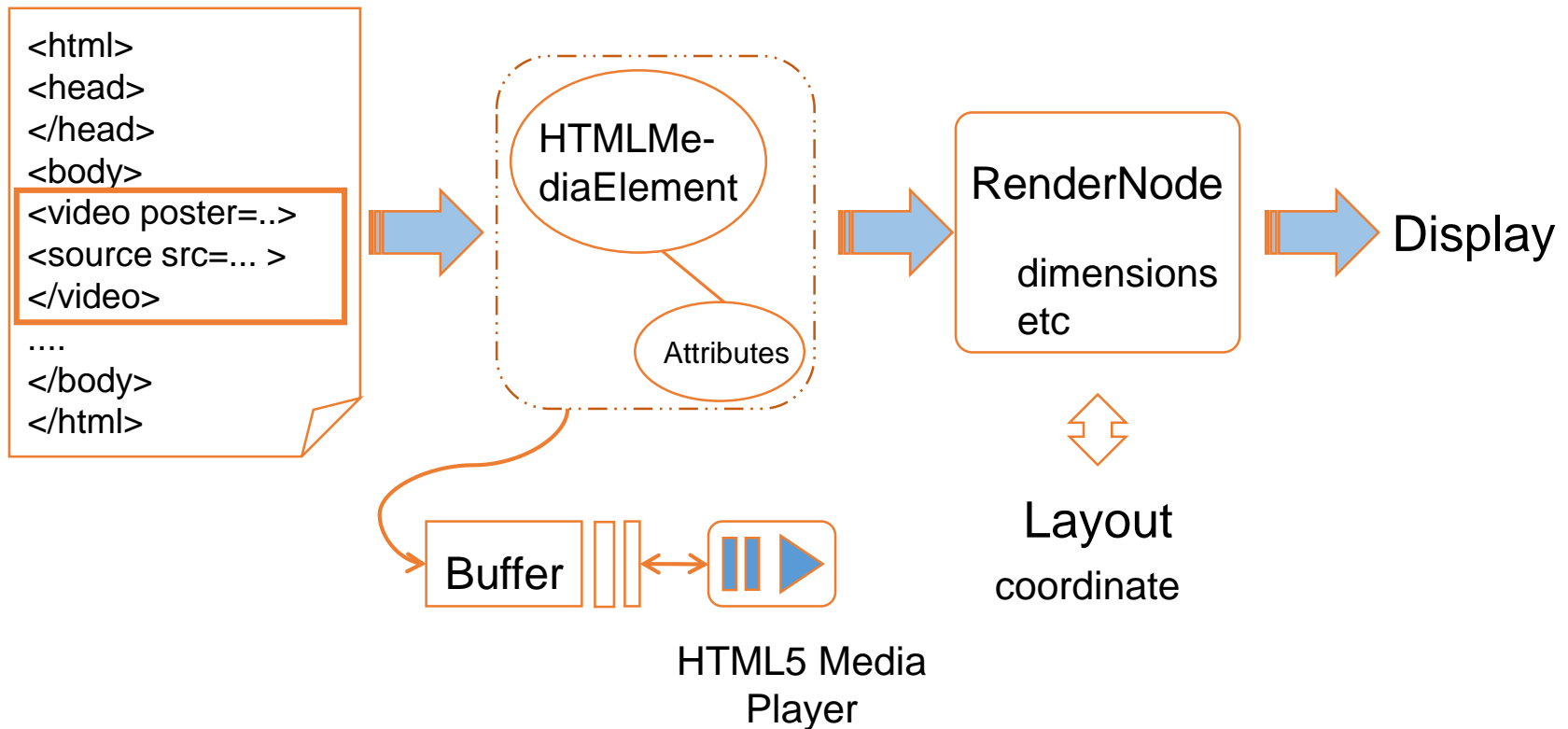
dengxi15@otcaix.iscas.ac.cn

HTML5: Structuring and Presenting content on WWW

HTML5: standard for web page programming language.

New requirement for browsers: support the multimedia natively (instead of by other third-party plugins formerly).

Browser's Processing Process

```
<html>
<head>
</head>
<body>
<video poster=..>
<source src=... >
</video>
....
</body>
</html>
```

HTMLMe-
diaElement

Attributes

RenderNode

dimensions
etc

Display

Buffer

HTML5 Media
Player

Layout

coordinate

Motivation

Existing testing works:

● Regression testing by browsers' developers

● W3C HTML5 test suite (15,000+ test cases).
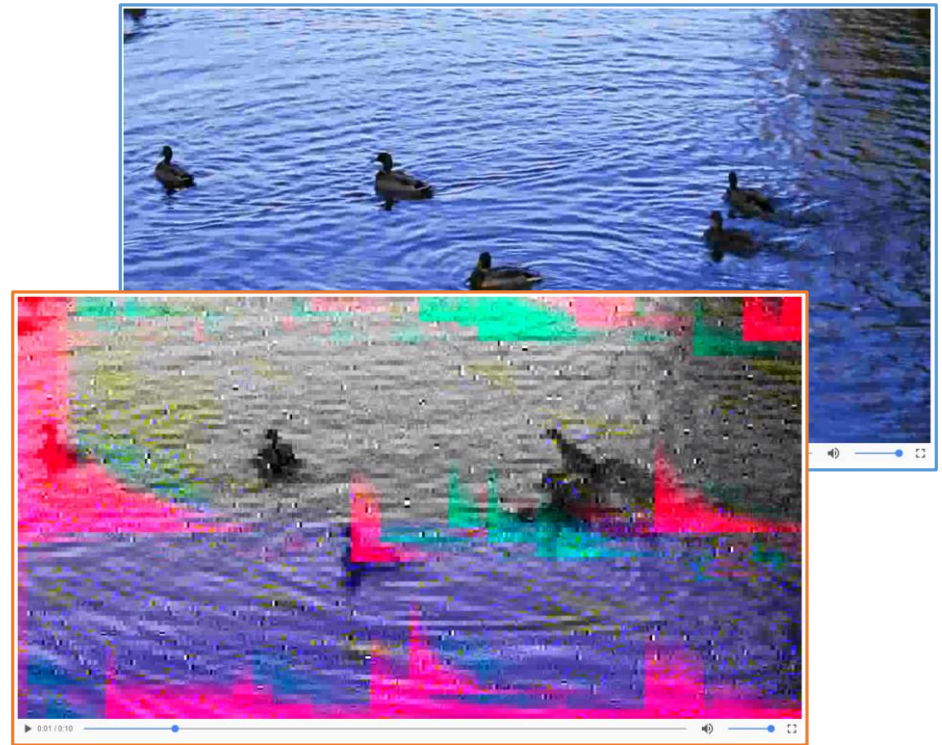
focusing on functionality of single points.

Motivation

Feature of HTML:

✎ multiple elements with various attributes with even various attribute values; referring to images and multimedia resources

🌐💻 tolerance of improper attribute value settings; recovering from improper transferring of files

Example: Chrome is playing an Ogg video file whose frame rate is 50fps and the chroma sub-sampling rate mismatches with its header

Easy to be peveived

Can browsers deal with complex conditions?

Combinatorial Technique

elements	media files
attributes	media file structure
attribute values	media properties
…	information in headers
	…

Our Approach

- **Model of CT**

- Test Generation

- Testing on benchmarks

Test Target

CT Models

Test Suites

Media ▶ Files

Web Pages

Model of CT

Visual contents' layout:

VideoCF: {MP4, WebM, Ogv}
PosterFormat : {PNG, JPEG, GIF, BMP, None}
WR, HR, PWR, PHR: {0.1, 1, 10, None}
Constraints:
for poster image

Properly displayed?

Resource loading:

ELement: {audio, video}
ElementSrc: {http, https, ftp, file, relative, not_set}
SourceSrc: {http, https, ftp, file, relative}
CROS: {anonymous, use-credentials, not_set}
TYpe:{audio/mp3, ..., video/mp4,...}
ContainerFormat:{MP3, AAC, Oga, MP4, WebM, Ogv}
Constraints:
$(ES \neq not\_set) \rightarrow ((SS = \#) \wedge (TY = \#))$
$((ES=relative) \vee (SS=relative)) \rightarrow CO = not\_set$

Can load?
Can decode?

Playback of audio:

AudioCF: {MP3, AAC, Oga}
SampleRate : {8,24,48,96}
ChannelNumber: {mono, stereo, 5.1}
FrameSR, FCN: {<,=, >}
BitrateMode: {VBR, CBR}
Constraints:
according to container formats

Normally played?

Playback of video:

VCF: {MP4, WebM, Ogv}
FrameRate : {1, 25, 50}
ChromaSubSampling: {4:4：4, 4:2:2, 4:2:0}
SSR, SCN, SFR, SCSS: {<,=, >}
Constraints:
according to container formats

Normally played?

Model of CT

Resource loading:

ELement: {audio, video}
ElementSrc: {http, https, ftp, file, relative, not_set}
SourceSrc: {http, https, ftp, file, relative}
CROS: {anonymous, use-credentials, not_set}
TYpe:{audio/mp3, ..., video/mp4,...}
ContainerFormat:{MP3, AAC, Oga, MP4, WebM, Ogv}
Constraints:
$(ES \neq not\_set) \rightarrow ((SS = \#) \wedge (TY = \#))$
$((ES=relative) \vee (SS=relative)) \rightarrow CO = not\_set$

Can load?
•loading process may be affected by the network-related attributes;

Can decode?
•the specified 'type' attribute may result in the media file be passed to the wrong decoder (the program that plays the media)

Test case buildng:
•generate media files with various formats
•build the Ftp, Http and Https websites, and deploy media files on websites
•build the media elements inside an html web page

Model of CT

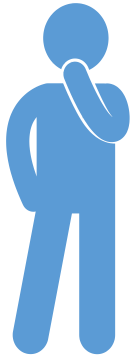## 📍 Visual contents' layout:

VideoCF: {MP4, WebM, Ogv}
PosterFormat : {PNG, JPEG, GIF, BMP, None}
**WR, HR, PWR, PHR**: {0.1, 1, 10, None}
Constraints:
for poster image

Properly displayed?
• the size information of element attributes ("width" and "height" attributes), the poster image (image file that indicated by the "poster" attribute), and video object may not cooperate with each other

Abstract parameters and values imply the degree of variation against that index of the video file

Model of CT
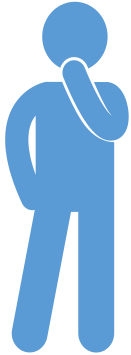
Normally played?
- the file with certain properties may impede the playback
- the properties indicated by the frame header or file header might be faulty

📍 **Playback of video**:

> **VCF:** {MP4, WebM, Ogv}
> **FrameRate** : {1, 25, 50}
> **ChromaSubSampling:** {4:4: 4, 4:2:2, 4:2:0}
> **SSR, SCN, SFR, SCSS**: {<,=, >}
> Constraints:
> according to container formats

📍 **Playback of audio**:

> **AudioCF:** {MP3, AAC, Oga}
> **SampleRate** : {8,24,48,96}
> **ChannelNumber:** {mono, stereo, 5.1}
> **FrameSR, FCN**: {<,=, >}
> **BitrateMode:** {VBR, CBR}
> Constraints:
> according to container formats

Model of CT

Bits	1	2	...	11	...	17	...	20	21	22	..	25	26	...	32
Binary	1	1		1		1		0	0	0		0	1		
Meaning	MP3 Syncword					Bit Rate			Sample Rate			Channel			

MP3 Frame → MPEG Header / Audio Data
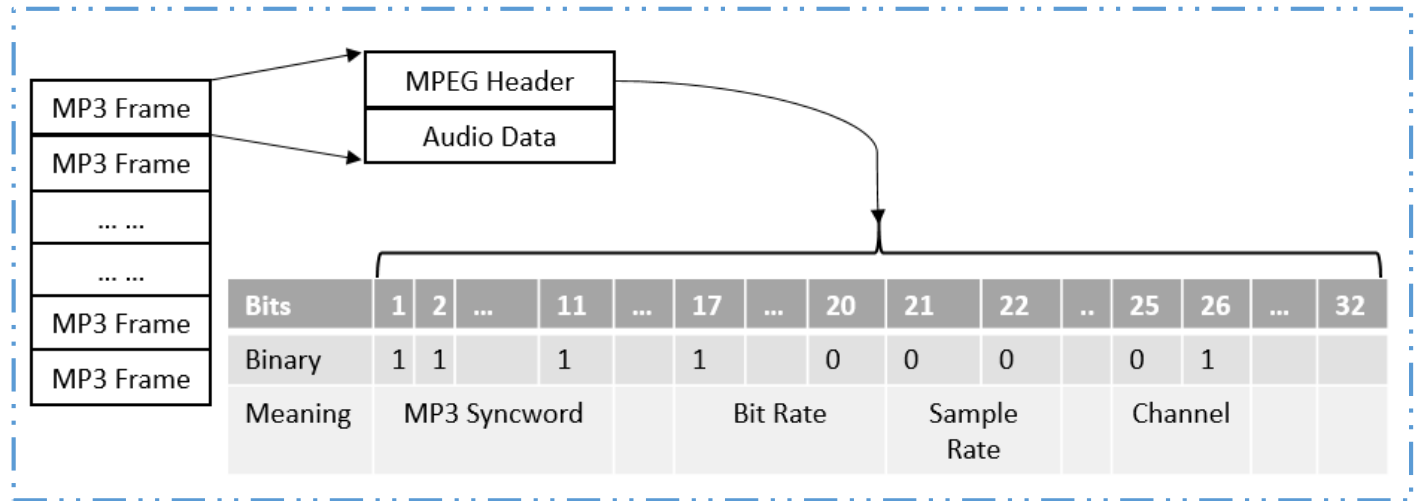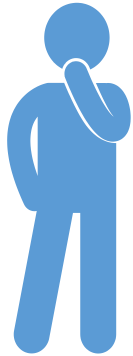
Test case buildng:
- generate the raw media data
- convert it to certain formats
- generate the header
- form a media file
- build an html web page with the audio element directing to this media file

Playback of audio:

AudioCF: {MP3, AAC, Oga}
SampleRate : {8,24,48,96}
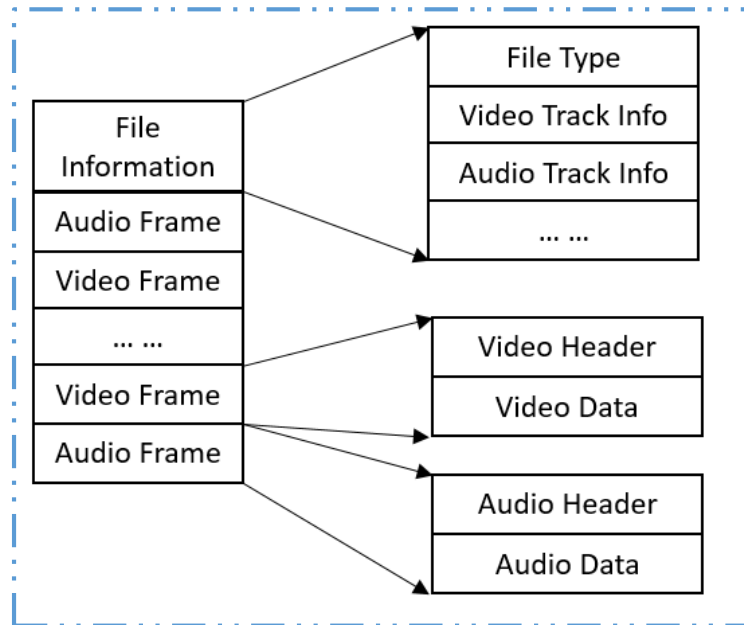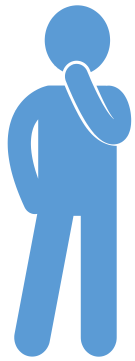**ChannelNumber:** {mono, stereo, 5.1}
FrameSR, FCN: {<,=, >}
BitrateMode: {VBR, CBR}
Constraints:
according to container formats

Model of CT

Playback of video:

VCF: {MP4, WebM, Ogv}
FrameRate : {1, 25, 50}
ChromaSubSampling: {4:4：4, 4:2:2, 4:2:0}
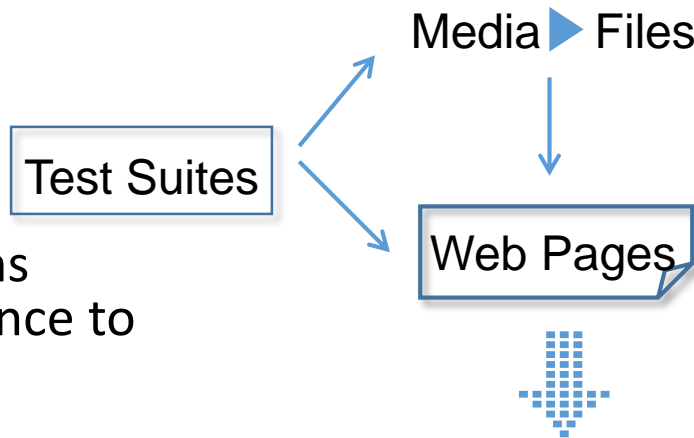**SSR, SCN, SFR, SCSS**: {<,=, >}
Constraints:
according to container formats

Experiment

Test cases:

162 web pages that contains media elements and reference to various multimedia files

Subject applications:

Horizontal analysis on 9 popular user agents

Vertical analysis on five historical versions of UC browser

Media ▶ Files
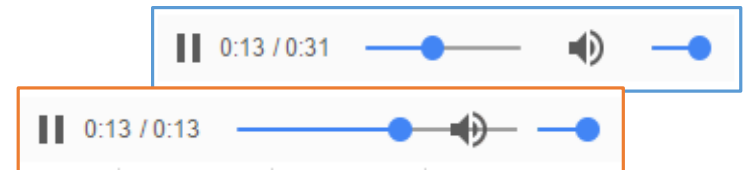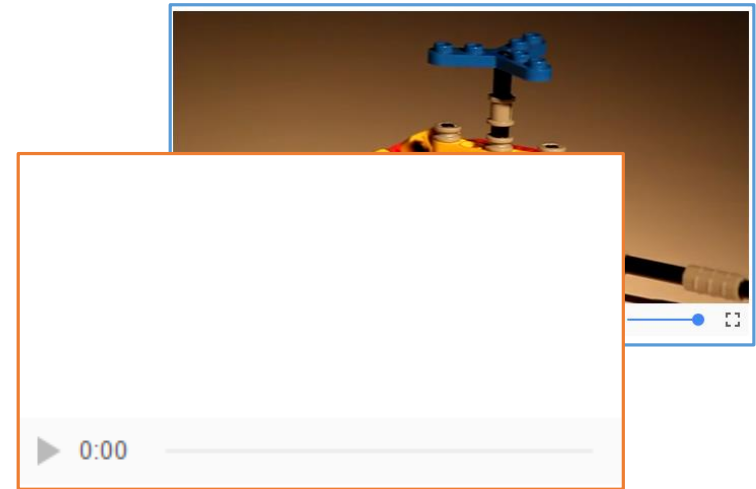
Test Suites

Web Pages

Observations -- Horizontal

📍 Resource Loading:

Failed fetching of data with CROS under certain URL schemes

- eg. URL = ftp, CO ≠ not_set
- Result: Firefox, Chrome etc. can't load this file

Lack of robustness for mismatched audio coding format

- eg. ES = not_set, CF in {AAC, MP3, MP4}, Type in {audio/ogg, video/ogg, video/webm}
- Result: IE, Opera, Edge etc. can't play this file

Vulnerable loading process with FTP protocol

- eg. URL = ftp, CF in {Oga, Ogv, MP4}, poor network circumstance
- Result: Chrome displays this file with wrong controls

Observations -- Horizontal

Visual Contents' display:

Wrongly calculated size of playback area

- eg. PF ≠ None, WR = None, PWR/PHR ≠ 1
- Result: ratio = 1 ≠ PWR/PHR

Disproportionately re-sized images

- eg. PF = GIF, WR/HR ≠ PWR/PHR
- Result: the poster image is re-sized to fit the playback area

Bad visual experience →

- eg. WR = 0.1, WH = 1
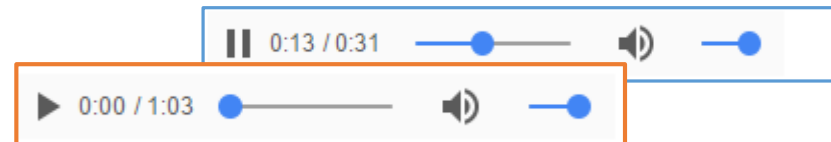- Result: pillar-boxed display of video and compressed controls

Observations -- Horizontal

📍 Playback of audio objects:

Wrong duration displayed

•eg. ACF = MP3, BitMode = CBR, FSR = less

•Result: longer duration displayed in Opera, IE and Edge.

Accelerated or decelerated playback

•eg. ACF = Oga, FSR = more

•Result: speeded up sound, with shorter duration displayed in Opera, Chrome and UC.

Distorted sound

•eg. ACF = Oga, CN ≠ stereo, FSR ≠ eq

•Result: distorted sound in Opera, Chrome and UC

Observations -- Horizontal

📍 Playback of video objects:

Inadequate support for high chroma sub-sampling rate files

- eg. VCF = MP4, CSS ≠ 4:2:0
- Result: video is played with no video image in Firefox, Opera, IE and Edge.
- eg. VCF = WebM, CSS = 4:4:4, SCSS ≠ eq
- Result: web page crashes in Opera and Chrome

Abnormal playback for certain frame rates and mismatches of video features

- freezing
- macro-blocking
  - eg. VCF = Ogv, FR = 25, SCSS ≠ eq
  - Result: video image is messed up in Opera, Chrome and UC
- accelerating or decelerating

Observations -- Vertical

→ Bugs may last in multiple versions.

- eg. Bug with id 6
- ACF = Oga, FSR ≠ eq
- Result: The playback of sound is accelerated or decelerated.

New bugs get introduced in newer releases.

- eg. Bug with id 3
- EL = audio, URL = https, CF in {Ogv, AAC}
- Being brought in with the extending of supported formats

Occurrences of bugs in historial vesions of UC
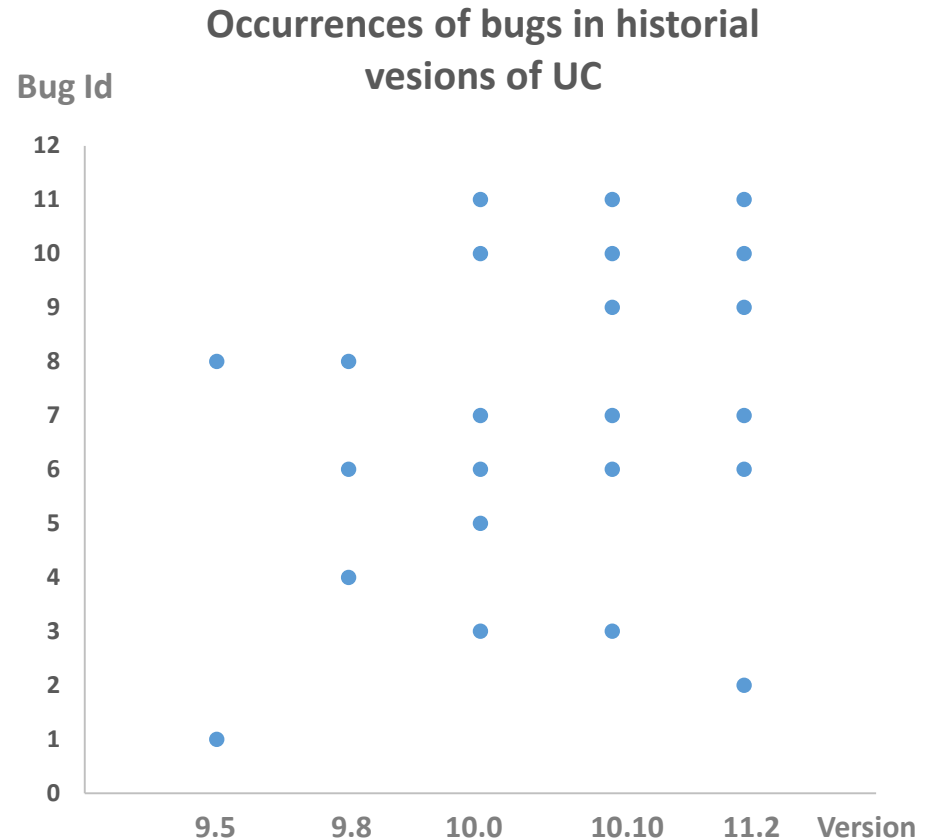
Conclusion

Demonstrate the effectiveness of CT

👆 Findings:
- defects exist widely among current popular browsers
- occurrence of bugs is unpredictable and inevitable if with no systematic testing

Thank you!