
M.Sc. Applied Computing

Project Proposal

A framework for deprecating
Flash in favour of HTML5
video in a web-based live
video streaming application

Luke Jenkins (p119783)

Contents

Contents	1
1 - Introduction	2
2 - Background	3
Flash	3
HTML5	4
3 - Aims & Objectives	7
4 - Research & Development Methods	8
5 - Resources & Planning	11
6 - References	13
7 - Appendices	15
7.1 - Poster	15
7.2 - Presentation	16

1 - Introduction

The following proposal sets out a research project to design and develop a framework for replacing Adobe Flash Player with HTML5 Video in a web-based video streaming application. The product owner has specified that the replacement is a necessity for the business going forward and that the implementation of HTML5 and removal of Flash must be staggered.

SureView Systems International is a software development company based in Swansea. They currently produce a web-based security software application called Immix Cloud.

Immix is an application that is designed to bring security devices of varying manufacturers, based in a range of locations, together into one manageable interface. This means an alarm from one device can trigger an action on another device such as to record video or audio. One of the primary functions of Immix is to stream video from cameras to operators, within the web browser.

Immix has a custom Flash control that is used to play streamed video and audio, but also to send commands out to devices, such as triggering alarms and unlocking doors. This Flash control is central to the entire alarm processing functionality of Immix and could be considered the most important part of the application.

One of the issues facing SureView is that there is a growing customer demand to remove Flash from the application, due to its reputation as being bug ridden and insecure. Unfortunately, HTML5 Video has only recently become a mature enough technology to be able to match the functionality provided by Flash. Immix is an application for security monitoring therefore using a known insecure technology such as Flash reflects badly on the image of the company and also affects the confidence the customers have in the product.

The company is also looking to simplify its mobile offering and instead offer the full web application on mobile devices. The current problem with this is that Flash is not supported on either iOS or Android, two of the most popular mobile operating systems [25].

2 - Background

With Flash approaching the end of its life and HTML5 becoming a more and more viable replacement, a brief review into the current issues with Flash and HTML5 was completed along with the new functionality HTML5 is able to offer.

Flash

Adobe Flash Player is a widely used web based plugin that allows websites to include interactive content in their web pages, including; live video, audio and 2D games. Over the years Flash has positioned itself as the defacto standard for interactive web content, though this position of dominance has been changing recently. With new security issues being discovered on a near daily basis and its track record for poor performance and device support, Flash is quickly becoming more and more undesirable as newer technologies begin to emerge.

Flash has a number of vulnerabilities that are being exploited more and more after becoming a regular target for malicious hackers [6]. There are currently over 1000 known security vulnerabilities ranging from simple methods of bypassing security controls and corrupting memory to methods allowing the execution of custom code [2] Although these issues are patched fairly quickly, the constant stream of new bugs and zero day vulnerabilities is one of the reasons manufacturers and developers are gradually dropping support for Flash in favour of newer, more secure alternatives[7].

One of the most recent vulnerabilities identified was the CVE-2017-3106 exploit which could allow the execution of arbitrary code [4]. Although, like many of the major vulnerabilities found in Flash, Adobe very quickly issued a fix, it was the fact the issue existed to begin with that is helping to gradually decrease confidence in Flash as a secure and reliable platform for multimedia content.

Not all devices support flash and general support is becoming less and less common, primarily due to the security and performance issues [3]. Most mobile devices no longer officially support Flash. iOS devices do not support it at all and the Flash Player has been removed from the Google Play store meaning Android device users can no longer easily install it [10].

Although Flash has essentially been superseded by HTML5, major companies and video streaming services still rely on it to this day [3].

A common complaint about Flash is that it performs so poorly that it often puts so much strain on a system that it is able to massively affect overall performance and can dramatically decrease battery life in mobile devices. The primary reason for this due to major inefficiencies in it's code, causing unnecessary over allocation and overuse of system resources [1].

Tests have revealed that using Flash can slow your PC down by as much as 80% [11]. Mark Hachman, a senior editor at PCWorld (Magazine) conducted tests using Flash with all the major browsers.

HTML5

Flash has long been a leader in providing an interface for developers to produce interactive and creative content to web users [12]. With Adobe finally looking to end-of-life Flash before the end of 2020, HTML5 becomes the new default choice when producing interactive content for the web [12]. HTML5 itself is able to provide the same video and audio functionality as Flash, though it does require additional functionality provided by CSS and Javascript to produce the interactive functionality that Flash is so well known for [12].

HTML5 introduced a new audio DOM object that allows audio files and streams to be played directly in the browser. This means audio can be played without the user needing to install additional plugins or extensions. Unfortunately the codecs supported by the audio DOM object are browser dependant and therefore different browsers will inevitably support different audio codecs.

A new video DOM object was also introduced which provides the exact same functionality as the audio DOM object but for videos instead. Controls relating to the video played can be styled using CSS and controlled using Javascript. The new video object not only supports the majority of the functionality provided by Flash's video player, but also includes additional functionality. One such additional piece of functionality is the ability to adjust the playback rate of the video. This means video can be played faster or slower depending on the users' requirements. Flash is

able to do this also but not without updated media headers being sent from the video server (RTMP Server) [17].

The HTML5 canvas is the ideal replacement for Flash's interactive content functionality, providing support for animations, transitions and games. The canvas is a defined area within a HTML page that allows scriptable manipulation and rendering of images to take place [15]. JavaScript functions are used to manipulate the contents of the canvas to suit the users needs. This functionality could be used to create interactive cross-platform games, dynamic graphs, and 2D animations [16].

Example usage of the canvas element is as part of a HTML5 game development framework by Phaser [20] The framework allows game developers to utilise the power of the HTML5 Canvas element and the WebGL library without having to spend time learning and potentially unnecessarily fully understanding the mechanics of what the Canvas element is doing. This means developers of all skill levels will be able to create HTML5 games with greater ease and in a shorter space of time. This contrasts with Flash where there is a relatively steep learning curve when it comes to ActionScript development, though development frameworks do also exist for Flash [21].

Although HTML5 is considered the replacement for Flash, it is not without its limitations. One of these limitations is that audio and video decoding capabilities are browser dependant and there is currently no requirement for browser manufacturers to favour any one particular audio or video codec. Manufacturers are under no obligation to use the same media codecs and therefore cross-compatibility can be a serious issue, especially if the service providing the multimedia in question is only able to stream in a single format.

Browser dependant implementations of HTML5 can also cause issues as some browsers may introduce specific functionality that an application then gets built upon. This functionality may not exist in the HTML5 specification and therefore may not be provided by all browser manufacturers. This can mean code will have to be re-written or alternate versions will need to be made for different browsers, adding time, complexity, and reduced flexibility to the finished product. An example of this is the WebUSB API which allows the browser to communicate with USB devices on the user's machine. Currently, this functionality is only supported in Google Chrome [18].

Another issue is the maturity of the HTML5 standard. Although it was started back in 2012 [14], many of the more advanced functionalities were not added or expanded upon until later versions. Audio and video streaming support has only recently been improved to the point where it is actually viable in a commercial environment [19]. As an example Media Source Extensions are now supported across the most popular browsers but are still unsupported on mobile devices. HLS support however, is available on mobile devices therefore it is technically possible to produce a fully HTML5 compatible, cross-browser, cross-device video player, albeit with some device specific functionality being included [19].

3 - Aims & Objectives

The main aim of this research project will be to better understand the limitations and issues of replacing Adobe Flash Player in a web based video streaming application with HTML5 video and also to develop a framework for the replacement, over a period of time.

The following objectives have been identified:

- Investigate and research the current feasibility of HTML5 Video live streaming in a production environment.
- Produce a thorough implementation plan and framework for replacing Flash with HTML5 Video in a web application.
- Design and develop a HTML and JavaScript based framework to be used to allow Flash to continue to be used whilst HTML5 Video is also implemented.

On completion of this research project, two deliverables are expected to be produced.

The first deliverable will be a framework for replacing Flash with HTML5 video, detailing how the replacement will occur, how the staggered integration of functionality will occur, and also when each part of the process will take place. This framework must allow enough time for SureViews legacy integrations to be updated to work with HTML5.

The second deliverable will be custom JavaScript and HTML code that will provide a web application with the functionality to continue to support Flash for legacy operations or functionality currently unsupported by HTML5 video whilst it is in the process of being implemented. This will mean a web application that currently only supports Flash could slowly begin to offer HTML5 alternatives as they become available.

4 - Research & Development Methods

A deductive approach will be taken with this research project. The reason this approach will be taken is because HTML5 should be able to provide the same level of functionality as Flash, it will just need to be proven. This approach will be necessary for this project as the consensus may not often be the whole truth therefore if it is widely accepted that HTML5 is an adequate replacement for Flash then there will be evidence or information about previous projects that would back that statement up.

In order to validate the literature reviewed, the research strategy will include the examination of case studies of previous Flash to HTML5 projects. One of the benefits of reviewing case studies is that it allows examination of the required data within a specific context [24]. They provide real world examples of the usage of the topic being research including what happened, what went right and what went wrong.

The completed framework will be implemented using the Agile software development method Scrum. Tasks will be allocated and deadlines set based upon the timeline described in the framework.

One of the benefits of using scrum is increased quality. Continuous feedback is a big part of the scrum methodology [22] therefore, regular meetings with the product owner and teammates working on other aspects of the project could lead to valuable feedback that will greatly improve the quality of the final deliverable.

Another benefit of using scrum is Increased project control. Scrum requires that there are regular meetings to discuss project progress, these are usually daily stand up meetings [23]. Any adjustments in priorities or project direction can be discussed at these meetings, giving the team much greater control over the project.

Although Scrum has many benefits when used for development, in this case it has been chosen as the development method primarily because SureView already use this for their development projects. It would be possible to choose a different method but this would simply cause unnecessary complications when scheduling tasks and wouldn't actually end up adding much to the completed project.

The framework requirements described in this proposal only represent a portion of the work involved in bringing HTML5 into the Immix platform. Additional developers will be working on other aspects such as the back-end video streaming services that will provide the H.264 encoded video. Developing using Scrum will allow tasks to be coordinated effectively to ensure that the work defined within this project proposal will not affect the other, related tasks being

completed by other members of the team.

Prototyping will also be used to ensure that the functionality that is intended to be utilised actually functions as expected. A proof of concept of a working HTML5 video player has already been created and will be used as a basis for developing the code based framework that will manage both the HTML5 and Flash players.

User requirements will be created to describe exactly what the framework will need to do and how it will work. These user requirements will be provided by the product owner, which in this case will be Paul Roberts, Senior Solutions Architect.

One of the main risks to the project could be the lack of available research material. As this is a fairly recent topic, focusing on a very specific area, it may be difficult to find research material that is relevant to the exact criteria described in this proposal. A result of this may be the inclusion of more web based sources that have far less academic weighting than a typical journal or paper.

There are no legal issues with regards to this research project specifically, though when the framework is implemented, great care will need to be taken to ensure customers making use of both the Flash and HTML5 functionality are not faced with any loss of service. SureView's clients are security monitoring companies and therefore the video that they are generally trying to view will be that of break ins or emergencies in progress. Should the system fail, legal action to reclaim losses could follow.

There will be no data or results as such to analyse at the end of the project, however, success of the project can be measured.

The first measurement of success is to compare the created framework with the user requirements specified by the project owner. If all the requirements have been met then the project can be deemed a success.

Another measurement of success could be against the aims and objectives of the project. If upon completion, the project has completed all its objectives and reached its overarching aim then it could be considered a success.

The third measurement of success will be if the code aspects of the framework actually work in the live environment. SureView has a quality assurance team that complete testing on the application before it is released to customers. The changes implemented as a result of this project will be thoroughly tested by them and any failures will be flagged up for review to be fixed.

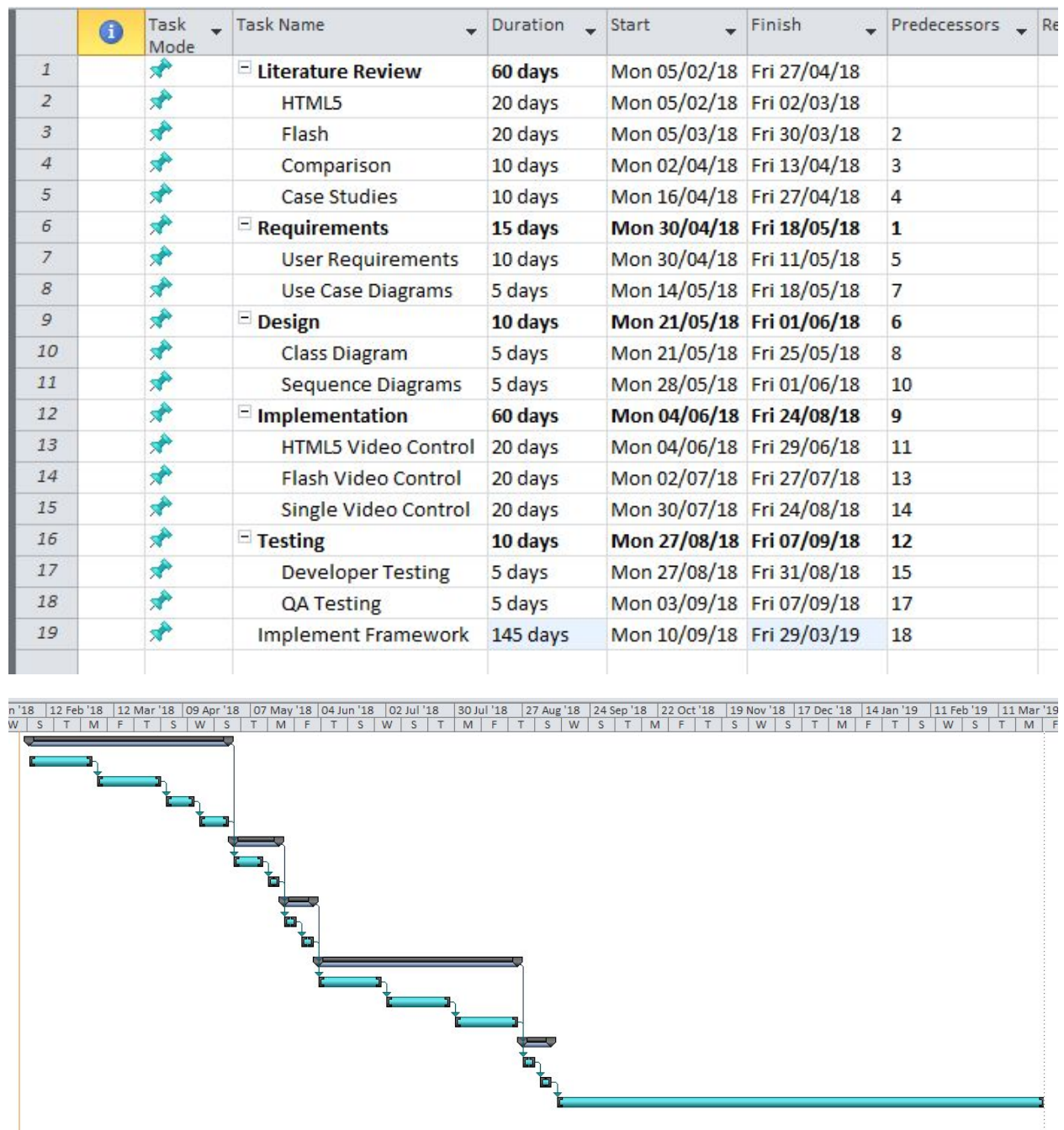
Specific testing documentation for the newly added features will be created for the QA team to use as a basis for their own testing activities.

5 - Resources & Planning

In order to complete this research project various resources will be needed to complete the research and actually implement the designed framework. The following resources will be required:

- **Development Machine** - A standard modern PC is all that is required, providing it is able to run the required applications.
- **Visual Studio** - This will be required due to the fact that the existing Immix Cloud project is an ASP.Net application, therefore the new HTML5 additions will be made using Visual Studio and merged into the existing project. The development will include use of the following programming/stylesheets languages:
 - C#
 - HTML
 - JavaScript
 - CSS
- **Flash Develop** - Although this project will focus on the addition of a HTML5 based video player, the existing Flash version will need amending so that it can work seamlessly with the new player.
- **Internet Access** - Access to the internet will be arguably the most important resource of this project as there is very little printed research into the subject.
- **Online Journals** - As above, online journals will be required in order to research other projects that have attempted to replace Flash with HTML5.
- **Library Access** - Access to the books in the university library will be required in order to research the subject area.
- **Online Library Access** - Access to online libraries such as Google Books will be required to research texts that are currently unavailable in the university library.

The following Gantt chart illustrates both the creation of the framework, and its implementation.



Additional resources not identified in this proposal may end up being required depending on any changes made to the project specification once it gets underway.

6 - References

1. **Reisinger, Don.** *How Security Flaws, HTML5 Sent Adobe Flash Into Decline.* eWeek, 2016.
2. **CVEDetails.com** Adobe Flash Vulnerability List. [Online] CVEDetails [Cited: 08 11 2017.]
https://www.cvedetails.com/vulnerability-list/vendor_id-53/product_id-6761/Adobe-Flash-Player.html
3. **Reinhardt, Robert.** *Is Flash Undead.* Streaming Media, 2015.
4. **Adobe.com** Adobe Security Bulletin, APSB17-23. [Online] Adobe [Cited: 08 11 2017.]
<https://helpx.adobe.com/security/products/flash-player/apsb17-23.html>
5. **Stevens, Luke. Owen, RJ.** *The Truth About HTML5.* Apress, 2014.
6. **Sarris, Simon.** *HTML5 Unleashed.* Sams Publishing, 2013.
7. **Freeman, Adam.** *The Definitive Guide to HTML5.* Apress, 2012.
8. **Pathan, Mukaddim Pathan. Sitaraman, Ramesh K. Robinson, Dom.** *Practical Systems for Live Streaming.* Wiley-IEEE Press, 2014.
9. **Santally, Mohammad Issack. Sungkur, Roopesh Kevin et al.** *Flash to HTML5 adaptations: A case-study for UpToTen.com.* IEEE, 2016.
10. **Salter, Anastasia. Murray, John.** *Flash and the Future.* MIT Press, 2014.
11. **PCWorld.com** *Tested: How Flash destroys your browser's performance.* [Online] PCWorld [Cited: 22 11 2017.]
<https://www.pcworld.com/article/2960741/browsers/tested-how-flash-destroys-your-browsers-performance.html>
12. **Adobe.com** *Flash & The Future of Interactive Content.* [Online] Adobe [Cited: 06 12 2017.] <https://theblog.adobe.com/adobe-flash-update>

13. **McLaughlin, Brett.** *What Is HTML5*. O'Reilly Media, Inc. 2011.
14. **W3.org** *HTML5.1 2nd Edition*. [Online] W3C [Cited: 06 12 2017.]
<https://www.w3.org/TR/html5/index.html>
15. **Fulton, Steve. Fulton, Jeff** *HTML5 Canvas: Native Interactivity and Animation for the Web*. O'Reilly Media, Inc. 2013.
16. **Hawkes, Rob** *Foundation HTML5 Canvas: For Games and Entertainment*. Apress. 2011.
17. **Sanders, William** *Learning Flash Media Server 2*. O'Reilly Media, Inc. 2007.
18. **Wicg.github.io** *WebUSB API: Editor's Draft, 24 September 2017*. [Online] Web Platform Incubator Community Group [Cited: 12 12 2017.] <https://wicg.github.io/webusb/>
19. **Jwplayer.com** *The State of HTML5 Video*. [Online] Photon Storm Ltd. [Cited: 12 12 2017.] <https://developer.jwplayer.com/articles/html5-report/>
20. **Phaser.io** *Desktop and Mobile HTML5 game framework*. [Online] Longtail Ad Solutions, Inc. [Cited: 10 01 2018.] <https://phaser.io/>
21. **Gamua.com** *Starling: The Cross Platform Game Engine*. [Online] Gamua GmbH. [Cited: 10 01 2018.] <https://gamua.com>
22. **Rubin, Kenneth S.** *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley. 2012.
23. **Shiel, James.** *The ScrumMaster Study Guide*. CRC Press. 2016.
24. **Zainal, Zaidah.** *Case study as a research method*. Faculty of Management and Human Resource Development, Universiti Teknologi Malaysia. 2007.
25. **Gelenbe, Erol. Lent, Ricardo.** *Information Sciences and Systems 2013: Proceedings of the 28th International Symposium on Computer and Information Sciences*. Springer Science & Business Media. 2013.

7 - Appendices

7.1 - Poster

A framework for deprecating Flash in favour of HTML5 video in a web-based live video streaming application

Luke Jenkins



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Abstract

Flash is a widely used web technology used to provide video streaming capabilities within websites and web applications. With more and more security issues arising and much more reluctance on the part of the users to use the technology, a replacement is desperately required. HTML5 has been named as the defacto replacement for Flash though has not been proven, at least at an enterprise level, to be able to provide or exceed the level of functionality that Flash could [1].

Flash

- A number of high profile vulnerabilities have been uncovered and more continue to be discovered on a daily basis.
- Not all devices support Flash and the most popular mobile operating systems iOS and Android no longer support it [2].
- Flash is incredibly inefficient and provides poor performance by over utilizing system resources.
- Product almost at the end of its life cycle. Maintaining a Flash application will become much more difficult once official support has been dropped [3].

References

1. Stevens, Luke. Owen, R.J. *The Truth About HTML5*. Apress, 2014.
2. Salter, Anastasia. Murray, John. *Flash and the Future*. MIT Press 2014.
3. Adobe.com *Flash & The Future of Interactive Content*. [Online] Adobe [Cited: 06 12 2017.] <https://theblog.adobe.com/adobe-flash-update>
4. Fulton, Steve. Fulton, Jeff *HTML5 Canvas: Native Interactivity and Animation for the Web*. O'Reilly Media, Inc. 2013.



Aims & Objectives

The purpose of this research project is to better understand the issues and limitations of updating the Adobe Flash Player with HTML5 Video in an enterprise grade video streaming system, and to develop a framework for a phased replacement over a limited period of time.

In order to achieve this aim, the following objectives will need to be met:

- Investigate and research the current feasibility of HTML5 Video live streaming in a production environment.
- Produce a thorough implementation plan and schedule for replacing Flash with HTML5 Video in a web application.
- Design and develop a HTML and JavaScript based framework to be used to allow Flash to continue to be used whilst HTML5 Video is also implemented.

HTML5

- Provides and embeddable DOM object that allows video data to be played within the web browser.
- Provides and embeddable DOM object that allows audio data to be played within the web browser.
- Provides and embeddable DOM object that allows graphical elements to be drawn and manipulated within the web browser [4].
- Relatively new standard and therefore has not been as widely implemented in Enterprise level applications.
- Inconsistent codec support across browsers.

Methodology

Case studies will be reviewed to validate the level of functionality provided by HTML5

A framework will be produced which will detail the exact steps required in order to implement HTML5 in place of Flash over a controlled period of time

Reusable JavaScript files will be created as part of this framework which will allow Flash to continue to function whilst the HTML5 features are being implemented and slowly added over time. These will be developed using the scrum development method using a series of smaller tasks over a defined period of time.

7.2 - Presentation

A framework for deprecating Flash in favour of HTML5 video in a web-based live video streaming application

Project Proposal

Luke Jenkins
luke-jenkins1@live.co.uk

- Project Background
- HTML5 vs Flash
- Aims & Objectives
- Methodology
- Expected Outcomes

Project Background

Immix Cloud is a web application that provides functionality to control and manage various pieces of security hardware from a single software solution.

Currently, Immix provides video streaming capabilities from cameras using Flash. Unfortunately, Flash is known to suffer from a number of security and performance issues and is also rapidly approaching the end of its life cycle.

HTML5 has been identified as a platform that should be able to provide the same level of functionality as Flash, within a web application.

HTML5 vs Flash

HTML5 has been designed to replace Flash as the method of choice for implementing video and audio streaming and advanced graphics manipulation in web browsers.

Flash currently has a number of issues relating to:

- Security
- Performance
- Compatibility
- Support

Aims and Objectives

The main aim of this research project is to identify the issues and limitations of updating the Adobe Flash Player with HTML5 Video in an enterprise grade video streaming system, and to develop a framework for a phased replacement over a limited period of time.

In order to achieve this aim, the following objectives will need to be met:

- Investigate and research the current feasibility of HTML5 Video live streaming in a production environment.
- Produce a framework for replacing Flash with HTML5 Video in a web application.
- Design and develop a HTML and JavaScript based functionality to be used to allow Flash to continue to be used whilst HTML5 Video is also implemented.

Methodology

A deductive approach will be taken with this research project. HTML5 should be able to provide the same level of functionality as Flash, though this would need to be proven.

In order to validate the literature reviewed, the research strategy will include the examination of case studies of previous Flash to HTML5 projects.

The completed framework will be implemented using the Agile software development method Scrum. Tasks will be allocated and deadlines set based upon the timeline described in the framework.

Expected Outcomes

It is expected that as a result of this project:

- A framework will be produced which will detail the exact steps required in order to implement HTML5 in place of Flash over a controlled period of time
- Reusable JavaScript files will be created as part of this framework which will allow Flash to continue to function whilst the HTML5 features are being implemented and slowly added over time.

Summary

- Research and identify any limitations of the HTML5 platform and potential methods of remediation.
- Using previous case studies and projects to identify potential pitfalls.
- Produce a framework for replacing Flash with HTML5
 - Produce JavaScript and HTML code files to provide Flash and HTML5 functionality during the transition
- Use Scrum to implement the completed framework

Questions?

