Technology Reports

Player QoE Quality Improvement

New Platform Technology to Further Improve the Quality of Multimedia Services —MediaSDK Software Library—

Communication Device Development Department Ginpei Okada Kazuki Asai Yunsang Oh

The multimedia service applications such as "dTV®*1, dTV channel®*2, dAnime store, and Hikari TV®*3 for docomo include a software library called MediaSDK which commonizes all functions required to provide services such as streaming processing. MediaSDK is a player required for video and audio data delivery, and because it has a significant impact on the quality of playback experienced by service users, its development significantly contributes to improving the quality of NTT DOCOMO's media services. This article describes an overview of MediaSDK and initiatives to improve the quality of related software.

1. Introduction

NTT DOCOMO has developed a number of services and applications respectively for the dramatically changing and highly competitive world of video delivery services, although there are issues with development efficiency. Hence, we developed MediaSDK (a software development kit) as common library with common functions for media applications (**Table 1**) to improve development efficiency, quality and maintainability.

NTT DOCOMO aims to improve the quality of MediaSDK from the perspectives of (1) the quality of the actual software, and (2) the Quality of Experience (QoE) of streaming*4. This article describes the developmental methods used to improve the quality of software, and analysis of quality data to improve QoE. The article also introduces issues and initiatives planned for the future.

2. Developmental Methods of MediaSDK

Currently, DOCOMO provides MediaSDK for the services below (**Table 2**). Since many services are provided through MediaSDK, the different

*1 dTV®: A trademark or registered trademark of NTT DOCOMO, INC.
*2 dTV channel®: A trademark or registered trademark of NTT DOCOMO, INC.
*3 Hikari TV®: A trademark or registered trademark of NTT Plala Inc.

Table 1 Overview of functions available with MediaSDK

Type	Details
Applicable media	Video, audio, subtitle
Delivery format	Streaming playback, local playback
Service model	Linear live playback, live catch-up playback, VoD
Delivery quality adjustment	Adaptive bitrate streaming, fixed bitrate streaming
Playback functions	Playback speed adjustment (0.5 to 32x), pause, seek, fast-forward/rewind
Supported devices	Android, Android TV, iOS/tvOS™, Web browsers
Digital rights management	Device-mounted DRM used (PlayReady, Widevine, FairPlay, etc.)
Codec	Hardware decoder used (H.264, H.265, etc.)
Data analysis	QoE data reporting function
API	APIs provided to application developers for each OS
Others	Device function linkage such as HDR, high-resolution audio

VoD: Video on Demand
tvOS™: A trademark of Apple Inc.

Table 2 Services (as of December 2018)

Service	Android	iOS	PC (HTML5)
dTV	○	—	—
dAnime store	○	—	—
dTV channel	○	○	○
DOCOMO TV Terminal® home app	○	—	—
Hikari TV for docomo	○	—	—
Other company's VoD service (service name not disclosed)	○	—	○

DOCOMO TV terminal®: A registered trademark of NTT DOCOMO INC.

hardware, operating systems, Digital Rights Management (DRM)*5 methods, encoding settings, and contents formats, etc. involved must be handled individually. There are also demands for regular releases of MediaSDK with the flexibility to handle a wide range of service requirements.

2.1 Cross-platform Support

MediaSDK consists of functions required for video and audio playback, and plug-ins to customize those functions for various services. Normally, development of a new application requires complex implementations on an OS. In contrast, MediaSDK modularizes these complex implementations, and

*4 Streaming: A communication method for sending and receiving audio and video data over the network, whereby data is received and played back simultaneously.
*5 DRM: Functions for protecting copyrights of digital content by restricting redistribution and preventing unauthorized copies, etc.

provides a commonized Application Programming Interface (API)*6, which makes it easier for developers to develop service applications.

Regarding plug-ins, our development must proceed efficiently on multiple platforms such as Android™*7, iOS*8 and PC, so the MediaSDK software includes plug-ins written in JavaScript*9 language for the cross-platform common logic section which enables commonization across operating systems (Figure 1).

Changing the software logic of each service can be done by updating the JavaScript section (Table 3). Furthermore, with modern Android and iOS smartphones, if conditions such as "primary purpose of the application is not changed," "a different storefront is not created," and "security is maintained" are satisfied, some sections of logic or parameters may be changed after delivery to the application stores (App Store®*10, Google Play™*11) by overwriting the script*12 (such as JavaScript) without altering the application itself. Therefore, a JavaScript plug-in hot patch can be delivered from the server to change the operation of an application using MediaSDK.

2.2 Agile Development Initiatives

MediaSDK entails development requests from various services received simultaneously and in

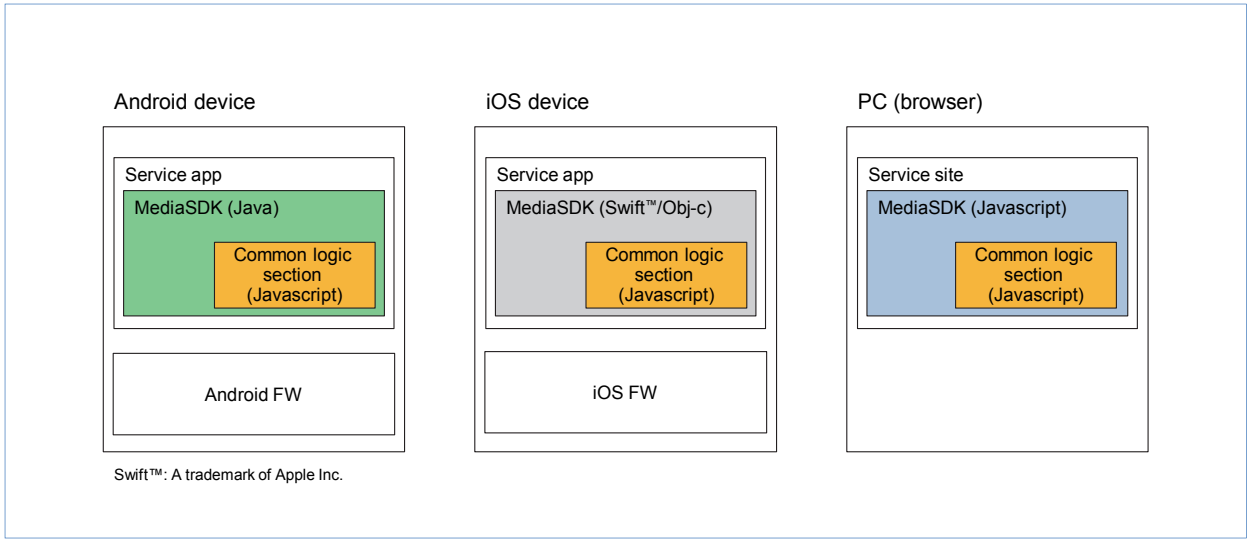

Figure 1 MediaSDK software structure

Table 3 JavaScript plug-ins

Plug-in type	Details
QoE plug-in	Sends QoE quality report and updates report data
ABR plug-in	Changes adaptive bitrate adjustment logic
Network plug-in	Changes destination content server connection
DRM plug-in	Changes DRM to use

*6 API: An interface that enables other software to use the functions available with an OS or middleware.
*7 Android™: A trademark or registered trademark of Google, LLC., in the United States.
*8 iOS: A trademark or registered trademark of Cisco Corp. in the U.S.A. and other countries, and used under license.
*9 JavaScript: A script (see *12) language appropriate for use in Web browsers. JavaScript is a registered trademark or trademark of Oracle Corporation, its subsidiaries and affiliates in the United States and other countries.
*10 App Store®: A trademark or registered trademark of Apple Inc. in the United States and other countries.

parallel as well as frequent requests for functions to be released, but because these cannot be properly handled with normal waterfall development*13 schemes, they are done with agile development*14 (Scrum development*15). In actual fact, the introduction of agile development schemes enabled 20 releases in FY 2017 including evaluation versions.

2.3 Testing and Release Automation

To achieve 20 releases per year, software must be evaluated rapidly. Time taken for manual testing will also detract from development. To solve these issues, we automated 93.2% of testing (2,750 items) to reduce the work load involved in evaluation.

We have also reduced the work load for creating release items (library, sample applications, porting guides, API specifications) by making automatic generation possible with a one-click linkage to Source Repository*16 systems such as Git*17. **Figure 2** shows an image of the automation of SDK release package creation.

2.4 Linkages to Functions in Devices (Decoder, DRM)

We have designed MediaSDK for playback using secure DRM methods and decoders supported by various operating systems and devices, and developed it with support for the DRM methods commonly used in recent years such as Widevine™*18/PlayReady®*19/FairPlay®*20, etc. so that services can be provided on a wide range of devices including PC browsers. For this reason, services such as those in **Table 4** are able to smoothly deliver high-quality audiovisual contents that require provision of high security level digital rights management technology.

However, particularly with Android terminals, depending on the handset model and chipsets*21 installed, there is a lot of variation in playback performance and quality which can affect the services users are experiencing.

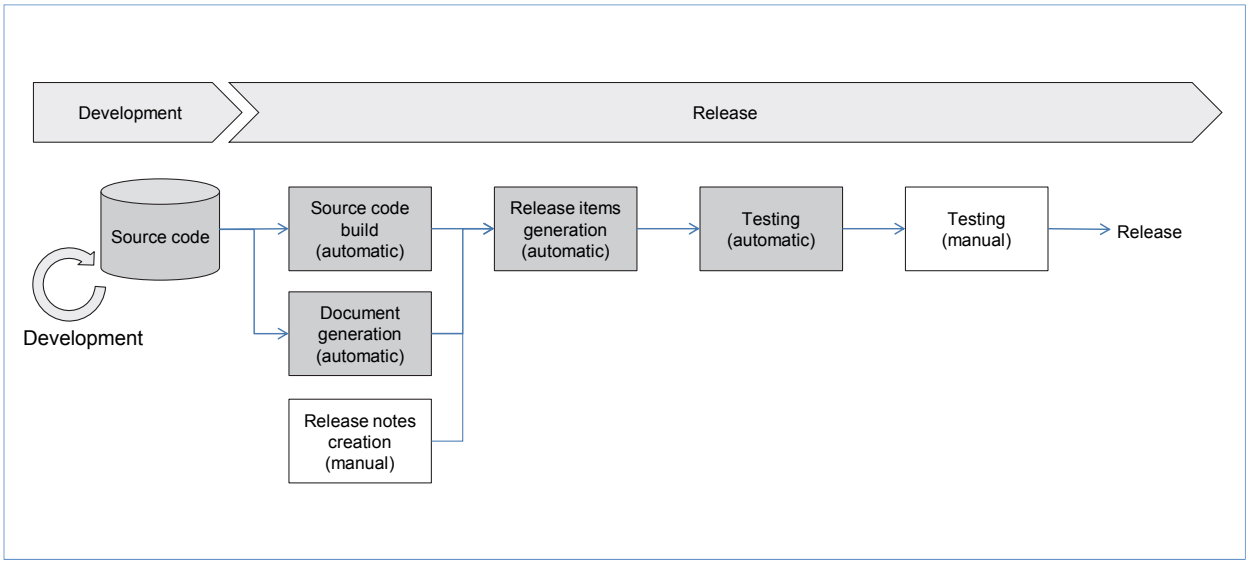

Figure 2 Image of automation of SDK release package creation

*11 Google Play™: A service from Google for delivering applications, video, music and books to Android terminals. Google Play™ is a trademark or registered trademark of Google, LLC. U.S.A.

*12 Script: A simple programming language for describing programs for simple processes. A program described by a script may also be called a script.

*13 Waterfall development: A development method in which the processes of definition of requirements, design, implementation and evaluation are performed in order.

*14 Agile development: A development methodology based on the Agile development declaration, a generic name for light development methods for rapid and adaptive software development.

3. Quality Data Analysis Initiatives

Apart from managing quality when developing software, there is also a mechanism in MediaSDK to further improve the quality of services after commercial release. The plug-in function that achieves quality management from the perspective of maintenance operations is described below.

3.1 Playback Quality Data Collection

Here, we describe initiatives to improve the quality of user experience with the functions of the QoE and Adaptive Bit Rate (ABR)*22 plug-ins. The former enables QoE improvement by building base for collection and analysis of data about video playback quality, while the latter enables QoE improvement through changes to its ABR logic.

Figure 3 shows an image of the operations of the QoE and ABR plug-ins, while **Table 5** shows examples of media playback quality data reported from the QoE plug-in.

The data from the QoE plug-in is in the JSON*23 format. In December 2018, approximately 30 GB of data (all Android smartphone users) were collected every 24 hours for the dAnime store. This data is analyzed to extract issues to improve playback quality, and then the extracted issues are reflected in the playback logic using the ABR plug-in to improve the playback quality.

Table 4 Example of applied high security level digital rights management technology

Service	Contents	Remarks
dTV	4K HDR10	
dTV channel	Dolby Atmos®	a-nation event live delivery on August 25 and 26, 2018
Hikari TV for docomo	Dolby Vision® HDR	

HDR: High Dynamic Range
Dolby Atmos®: Dolby Atmos and Dolby Vision are registered trademarks of Dolby Laboratories.

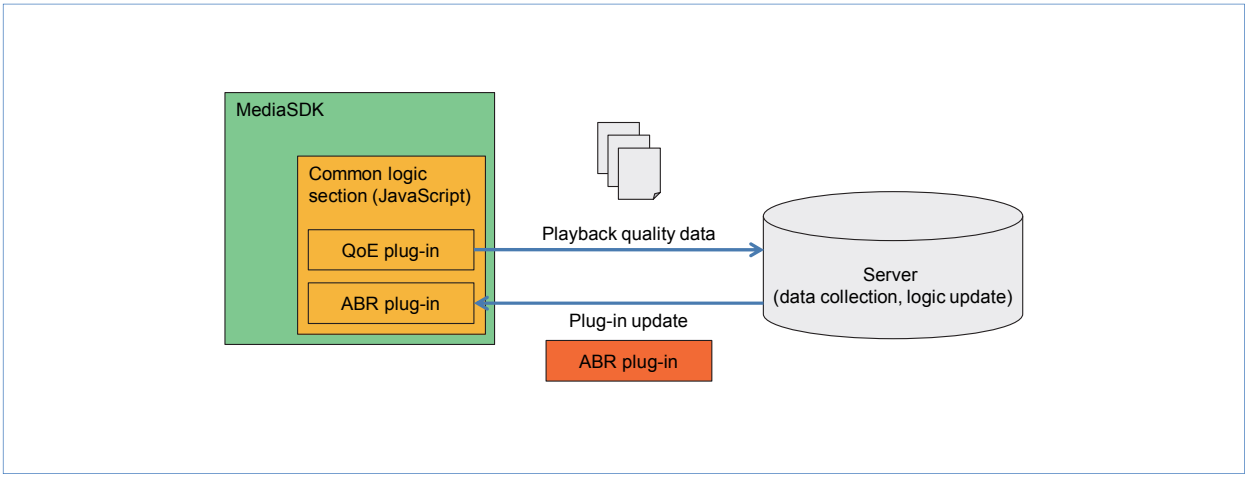

Figure 3 QoE, ABR plug-in operations system image

*15 Scrum development: An agile development method.
*16 Source Repository: An area where source code is stored.
*17 Git: A source code management tool.
*18 Widevine™: A trademark or registered trademark of Google, LLC., in the United States.
*19 PlayReady®: A trademark or registered trademark of Microsoft Corp. in the United States and other countries.
*20 FairPlay®: A trademark or registered trademark of Apple Inc. in the United States and other countries.
*21 Chipset: Devices that control mobile terminal software and various hardware processing. Devices such as the CPUs and control circuits are collectively referred to as "the chipset."

3.2 Playback Quality Data Analysis

We use the Plan, Do, Check, Act (PDCA) cycle*24 in MediaSDK maintenance operations. We describe the process of improving playback quality through quality data analysis based on PDCA (Table 6).

1) Planning

First, data reported from the QoE plug-in is modified in the server so that it can be visualized. Table 7 shows an example of visualization of the state of playback quality.

By analyzing visualized data, user QoE can be understood. The next step is to set the target values required for improvements. Main improvement targets include "time taken for playback to start," "time for re-buffering during playback," and "adjustment for selected ABR picture quality."

2) Doing

ABR logic is used to optimize playback by changing the picture quality to shorten buffering time. We create proposals in consideration of each case because although the ABR plug-in can be applied to all applications, it's also possible to apply customized logic for particular groups (of Internet Service Providers (ISP), device models, operating systems, etc.).

3) Checking

The created ABR plug-in is delivered to a specific user group, and a comparison is made of its quality with the existing logic (by A/B test). Because it's possible to perform the evaluation with the aforementioned visualization, the quality targets in Table 7 are used as the main evaluation items.

Table 5 Example of playback quality report data

Data	Details
Startup time	Time until playback of first video frame starts
Buffered duration	Cached continuous buffer time
Download time	Time to download a segment of video data
Size	Size of downloaded segment
Consecutive drops	Number of serial frame drops (value for measuring terminal performance)
Action	User event
Network errors	Network errors during playback
Others	Others, reporting for QoE-related data

Table 6 Maintenance operations with the PDCA cycle

Planning	Specific targets for improvements are set through data analysis with data visualization. Improvement proposals are studied from data analysis based on hypotheses.
Doing	An improved version of software is created based on the results of studies in the planning stage
Checking	A/B testing of the existing software and the improved version, and evaluation of results
Action (improvement)	Software is updated based on the results of the A/B testing

*22 ABR: Technology to dynamically change picture quality for playback to match communication speeds.

*23 JSON: A data description language based on object notation in JavaScript®.

*24 PDCA cycle: A method of ensuring smooth running of business. The PDCA cycle entails repeatedly and continually running through the four steps of (1) Plan (planning), (2) Do (performing), (3) Check (measuring results) and (4) Act (making improvements).

Figure 4 shows an image of the ABR plug-in A/B test

4) Action (improvements)

The optimal playback logic ABR plug-in is selected from the results of playback quality evaluation, which can be reflected in the player as required to improve quality. The PDCA cycle does not have to finish with one cycle but can be repeated to further improve quality.

3.3 Overall Evaluation of Playback Quality

The results of quality targets must be properly judged for their merits to advance the PDCA cycle for quality improvement by collecting and analyzing quality data, as mentioned. However, there are many quality targets that are variables in evaluations, and many variables in video playback are in trade-off relationships. Hence, there may be problematic cases where improving one target

Table 7 Playback quality visualization example

Playback quality target	Details
Average playback time	Average playback session (total playback time/total number of playback sessions)
Average buffer size	Size of data accumulated in device before playback
Number of error events, frequency	Error event trends
Start time	The time taken from the user's start playback operation to the actual start
Speed	The network environment throughput at the user side
Bitrate ratio	Trends of video and sound quality selected with ABR logic
Frame drop	Ratio of occurrence of data that cannot be processed due to the terminal performance

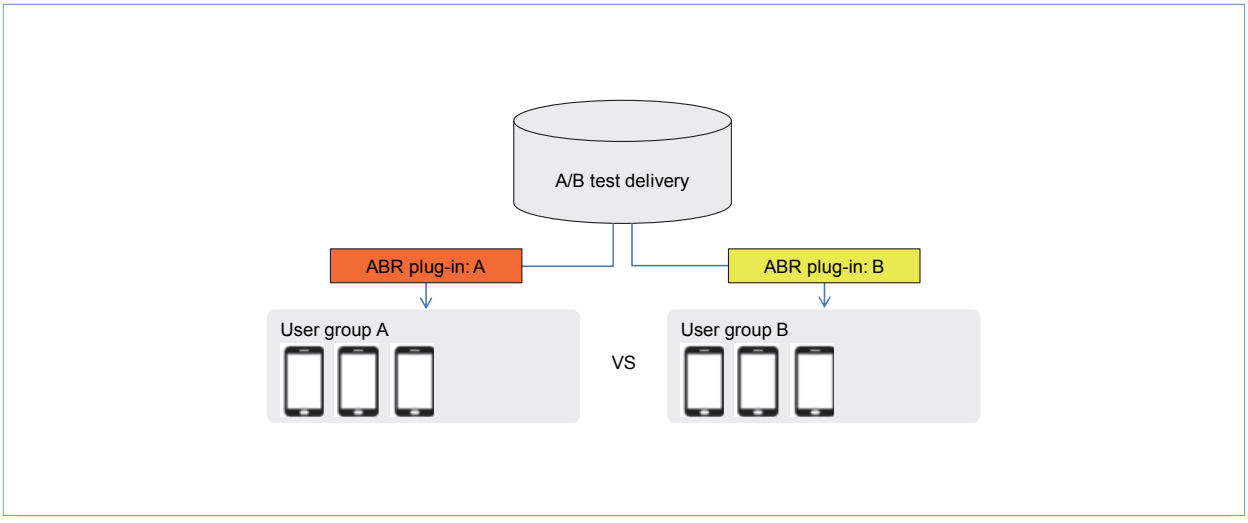

Figure 4 Image of the ABR plug-in A/B test

degrades another, making it difficult to judge merits fairly with A/B testing.

Hence, we use NTT Network Technology Laboratories' playback quality evaluation technology [1] to perform an overall evaluation. Using the technology described in [1], it's possible to calculate overall QoE values for points 1 to 5 using information about buffering with video playback or media picture quality. Comparing these QoE values makes it possible to maximize the levels of user satisfaction by maximizing the values through the PDCA cycle.

4. Issues

4.1 Differences in Device Performance

It's not always possible to make great improvements of streaming playback performance just by improving the ABR algorithm, because streaming playback performance is heavily dependent on device performance. Particularly with Android devices, where there is a lot of variation of the quality and performance of functions required for video services (decoding, DRM encoding processing, etc.) with each manufacturer, it can be difficult to apply uniform improvements. For this reason, we plan to study methods to measure device performance with playback and apply algorithms tailored for performance.

4.2 Overall System Optimization

This article has described improving playback quality, although QoE targets also depend on factors such as Content Delivery Network (CDN)*25, delivery servers and content encoding methods. Therefore, we intend to optimize overall service systems by repeating A/B testing with various combinations of servers, clients and contents.

4.3 PDCA Cycle Efficiency

The A/B testing and so forth we have discussed currently all require human operations. However, by promoting automated data analysis with the advances in quantitative scores such as QoE values and with AI technologies, we intend to make the PDCA cycle more efficient and put efforts into automating quality improvement of video delivery services.

5. Conclusion

This article has provided a general description of the development methods for the software library included in DOCOMO's multimedia service applications, and described how their quality is improved. We implemented agile development schemes for flexible cross-platform development. We also reduced working load and achieved greater efficiency by automating release. Going forward, we aim to further raise quality through repeated analysis, evaluation and improvements with the PDCA cycle using NTT Network Technology Laboratories' QoE visualization technology to analyze playback quality data.

REFERENCES

[1] K. Yamagashi and T. Hayashi: "Parametric Quality-Estimation Model for Adaptive-Bitrate-Streaming Services," IEEE Transactions on Multimedia, Vol.19, No.7, pp.1545-1557, Feb. 2017.

*25 CDN: A network solution optimized for fast and stable distribution of large files such as images and video.