Accelerating Multiscreen Video Delivery

By working with a seasoned systems integrator, organizations can more effectively and quickly spec, test and deploy multiscreen video with packaged or customer solutions that meet consumers' growing expectations for highly visual and engaging content.

Executive Summary

Today's video viewer seeks more and more options to consume entertainment content anywhere, anytime and on different devices. IP-based delivery of video content has made these options possible and, as a result, is dramatically influencing consumer viewing patterns.

From the days of traditional "broadcast" TV when carriers decided what and when consumers would watch, today's world has transitioned to a narrow, "me-cast" approach where consumers decide where, when and how they want to watch any piece of content. What ignited this? The ubiquity of the Internet, where content is now available anytime, anywhere, as well as the emergence of a plethora of transportable mobile devices that work across networks and content-serving platforms.

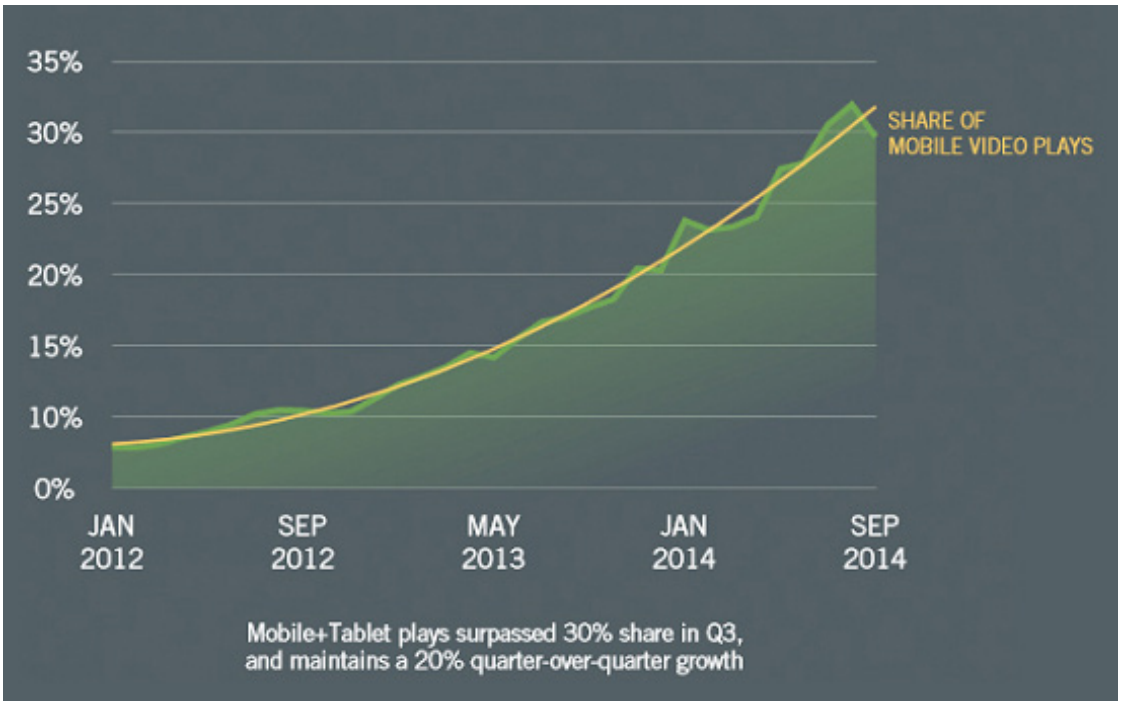
Figure 1 (next page) puts this trend into numerical context.

According to estimates from IHS (a research and analysis company), video consumption on set-top-boxes and mobile devices will be equal by 2015. As a result, delivering video content over multiple screens has become mandatory for carriers, content providers and distributors to stay relevant in the market. However, the challenges related to emerging technologies and evolving standards as well as operations and financial constraints often act as barriers to implementing multiscreen solutions.

This white paper discusses methodologies and processes that will help engineering to overcome these challenges. These methodologies have always existed in the software engineering world but this paper highlights how organizations across industries can successfully adapt them to build multiscreen solutions.

Cognizant

Mobile's Share of Online Video



Source: http://www.fierceonlinevideo.com/story/ooyala-smartphone-tablet-ott-viewing-jumps-114-last-year/2014-12-09
Figure 1

Engineering Challenges

Given the fundamental industry changes highlighted above, the engineering organization (be it within a carrier or a content provider) – the group ostensibly responsible for delivering content to consumers – faces unprecedented business and technology challenges. Delivering video over IP networks to multiple devices is fundamentally different from delivering video over established cable/satellite networks or with disk-based distribution. It is more akin to delivering software-based solutions and services over the Internet.

Given the rapid advancements in the Internet and mobile devices, the carriers and content producers need to innovate much faster than they have ever before. To increase their velocity, they must increasingly rely more on software-based innovation. At the same time, they cannot lose sight of goals related to operational efficiency and cost-effectiveness.

On one end, they must deal with new, Internet generation consumers who are demanding evermore services and viewing choices. With more options available, customers hold the upper hand and can opt out of services that do not live up

Balancing Cost Containment with Innovation

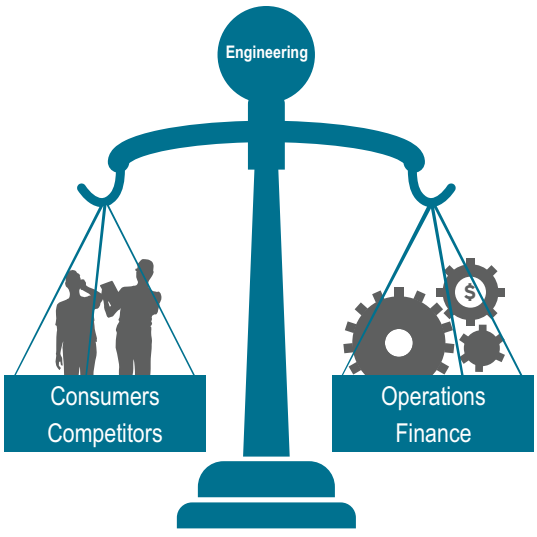


Figure 2

to their expectations. The demands around time-to-market and ease-of-use are too important to ignore. The competitive landscape is also very complex now. "Traditional" competi-

tors are dealing with homogenized products and services. Their only hope for competitive differentiation is through accelerated market delivery. Meanwhile, upstarts from a wide array of divergent e-businesses (think Apple, Google, Amazon) offer alternate products that offer more options for the consumers such as iTunes, Amazon Prime, etc. These new competitors have a different DNA – they have mastered software-based innovation to drastically reduce time-to-market and have a greater appetite for risk.

On the other end, the pressure to contain costs by industry CFOs remains in place. Engineering is expected to leverage existing investments already made in the infrastructure. For every new initiative, ROI needs to be established. And the solution needs to be implemented in such a way that operations can support it, with minimal overhead. These challenges are compounded by the evolving nature of still-maturing technologies, such as adaptive streaming, UHDTV, etc. and evolving CableLabs and SCTE standards.

Thus the engineering organization has to walk a tightrope – innovate quickly to retain market relevance and cash in on available revenue opportunities without overspending – very much like any software engineering house. If engineering doesn't innovate fast enough, it will lose customers for whom alternate offerings are just a click away. If they don't streamline by leveraging existing engineering investments, cost overruns could sink the business.

Multiscreen Solution Background

Before revealing how to address these challenges, let's examine the attributes of a multiscreen video delivery solution.

Figure 3 depicts the components of a multiscreen solution.

As illustrated in Figure 3, a multiscreen delivery platform is comprised of two distinct subsystems:

- **Video transformation subsystem:** This subsystem gets the content ready for delivery. It transforms the video content into the different forms and bit rates that are supported by multiple target devices. It encrypts the content using different digital rights management (DRM) technologies as supported over these devices and then packages and makes them available on the content delivery network (CDN).
- **Video delivery subsystem:** This subsystem takes over when a user requests content through a particular device. It ensures the authenticity of the user and the device requesting the content, and ensures the content is available and can be accessed by this user from his current location. It then directs the device to the appropriate content location on the CDN for playback.

Many of the components, especially in the video transformation subsystem, provide "commodity" services such as ingest, transcode, encrypt, etc. While these components provide the necessary

Anatomy of a Multiscreen Solution

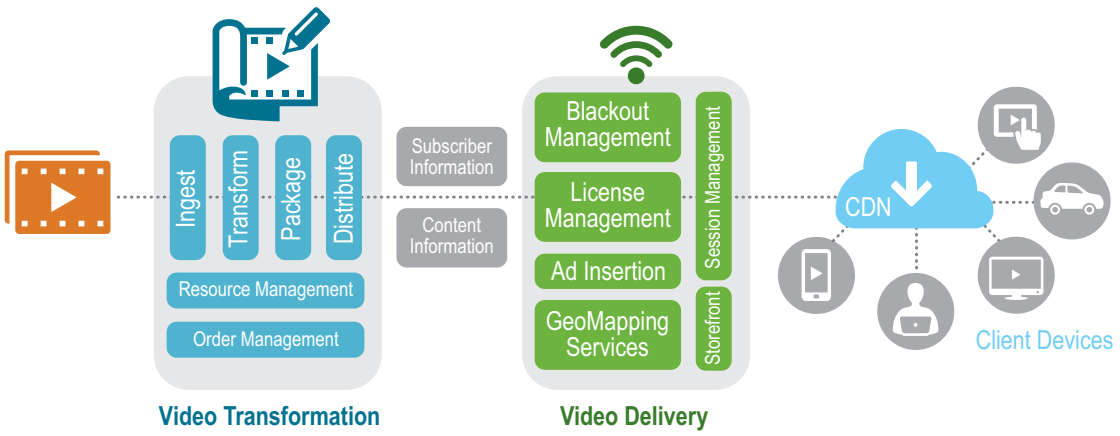


Figure 3

building blocks for the solution, they typically do not provide unique solution-specific value. However, these standard components provide an opportunity to control development costs. Hence, they are best procured from third-party providers. Established vendors offering numerous components include Aspera, which provides ingest/file transfer; Digital Rapids, Rhozet, etc. for transcoding; and Microsoft PlayReady and Motorola SecureMedia, which offer DRM for encryption, etc. Cloud-based options, such as thePlatform and Kaltura have also emerged and can dramatically reduce costs with their pay-per-use model. Of course, an organization can also host/integrate other components in the cloud to form a complete solution.

Components such as content metadata management, ad insertion, merchandising/cataloging for personalized experience, etc. offer the ability to inject differentiating features into your organization's multiscreen solutions. These components must be engineered very carefully since they offer the ability to create unique intellectual property (IP) that your organization can monetize. Third-party options are available for these components as well (e.g., mDialog, ThisTech for ad insertion), but build-vs.-buy decisions need to be carefully evaluated. Organizations need to strike the right balance between leveraging existing components to reduce time/cost while maintaining flexibility to customize the solution. Often, customizing available solutions to meet emerging market requirements is the most sensible approach.

Apart from these complexities, the platform would typically need to be integrated with many other operational components such as operations support systems (OSS), business support systems (BSS) and others.

While the platform for multiscreen delivery can be very complex, the challenges don't disappear on the device (i.e., client/presentation) side. The very purpose of a multiscreen solution is to provide the flexibility to the users to consume content on various kinds of devices like tablets, smart TVs, STBs, smartphones, gaming consoles, etc. The ecosystem is quite fragmented, with different devices supporting different technologies and specifications. On top of these, the user encounters different UI experiences when he moves from one device to another depending on the form factor. A viewing experience at a distance of 10 feet from the device for smart TVs, STBs, gaming consoles, etc. needs to be very different from the experience with devices like tablets or smartphones that typically have two-foot viewing distances. Hence, it is almost mandatory that your solution has a flexible and extensible UI platform that allows you to render the client UI and video on different devices without much overhead. While HTML5 can meet most needs of a UI on a browser-capable device, you need the UI framework that changes the layout, shows metadata based on the form factor and is flexible enough for enhancements to leverage performance of native players/apps.

Thus, on the platform side, a multiscreen solution results from careful build-vs.-buy decisions and efficient integration of various components and the other in-house systems such as OSS, BSS, archival, etc. On the client side, a systematic approach is required to develop a UI that renders equivalently on all devices, offering a pleasant, native user experience. This is not a one-time effort but an ongoing process, since most solutions evolve over time to address emergent user demands and devices. A strong solution lifecycle and efficient development processes are integral in enabling an end-to-end solution.

Systems Integration Requirements

This is where systems integrators can play a critical role. While the engineering department may have the necessary expertise and capacity to define the solution blueprint, they may not be able to see the long-term, ongoing program ramifications. Moreover, engineering is typically challenged when it comes to evaluating options, interacting with internal and external stakeholders, customizing third-party components and integrating them with custom-developed in-house components to form a complete solution. Lastly, engineering constantly contends with competing pressures for resources and capital infrastructure.

Engaging a systems integrator can enable engineering to maintain focus on other strategic initiatives. While engineering can control the strategic direction of the solution and maintain governance, the systems integrator can be engineering's single point of contact for all custom development, build-vs.-buy decisions, customization, liaison with third-party providers and integration. The systems integrator is also best equipped to play this role due to its varied skills and experience. Lastly, the systems integrator can also help with incremental development and release planning following the initial deployment of the solution.

To summarize, systems integrators should be considered by engineering to assist with multiscreen solution development for the following reasons:

- **They have expertise in most third-party solutions** that form integral parts of the solution. Since they have used different third-party products in varied situations, they know the pros and cons of these products and hence can provide valuable inputs during the evaluation phase.
- **They are experts in custom software solution development** to meet the requirements of the customer as well as customizing third-party products. Equally important, they can bring in the additional technical expertise that your engineering organization may lack.
- **They have user experience teams** with specific expertise in designing and developing client-side apps for different devices.
- **They have well-defined processes for custom development, software integration and automated testing** to guide the entire solution development and maintenance cycle. They have identified best practices for integration of custom-developed components with those procured from the vendor, and they perform integration testing in accredited lab environments.
- **They know how to complement your organization's resources and work as one team toward realizing your goals** in the event that timelines and efforts associated with multiscreen solution rollout exceed what engineering can commit to.
- **Most important, they have done it before.** They are typically engaged in building similar solutions for numerous customers. The learning that they can bring in the areas of tools and technologies, as well practices and processes, is too valuable to ignore.

The rollout of a multiscreen solution includes very complex content transform and delivery platform topologies to be designed, developed, tested, deployed, operated and renewed over time. These activities require experienced systems integration partners to help deliver the end result on time and within budget.

Agile Development

Dealing with evolving, ever-changing requirements and managing delivery velocity are other elements of a multiscreen solution development initiative. Consumers and end users drive only functional requirements of multiscreen solutions, which often change and evolve. But engineering teams often receive requirements not just from business teams that provide functional requirements but also from: operations teams that run/maintain the solution once it is deployed; third-party companies that provide solution components; security teams; and, at times, even the architecture team, which mandates certain requirements.

Since these requirements come in different forms and with varying priorities, following an Agile methodology provides a framework and discipline to funnel them into a well-defined, prioritized product backlog. This approach requires product owners (real or part-time) who provide respective solution requirements for the end solution. Since these stakeholders are not typically trained to provide the requirements in the Agile story forms that fit into the product backlog, it usually pays to have product owner proxies who can interface with product owners from different functions and capture their requirements in clear, concise and complete story forms. Product owner proxies can funnel these requirements into the backlog, taking into account the priority and clarity of each requirement. Thus, the dynamic requirements of a multiscreen solution can be managed systematically without overwhelming the development team.

The Scrum teams responsible for the development can work off product backlog and build the solution in iterative fashion. Sprint cycles of two to three weeks typically work well, providing the right balance between the development team's ability to develop and test the requirements within that timeframe and stakeholders' ability to change the course of the development at the desired pace. While it may not be possible to make a release to production at the end of every Sprint (especially while working on the first foundation version), Agile allows the engineering team to receive timely feedback from various stakeholders. Following Agile ensures early and frequent release of its solution.

The challenge in applying Agile to multiscreen solution development is often related to building custom components and customizing third-party components – while making sure that they can interoperate. In fact, it may even be counterintuitive to follow Agile to begin with when developing components that need to integrate with many others since application interfaces and contracts must be established.

For example, developing a session manager that receives a session request from the client side of the application before responding entails checking with the subscriber management system to verify the client side's validity and with the content management system to ensure the requested content exists, etc. The tendency may be to wait for the components to be available before developing dependent components to ease testing. But in reality, these interfaces change over time. The way to cope with this is to follow the Agile methodology.

Once the initial blueprint of the solution is defined and high-level interfaces among the components are agreed upon, development of all components can occur in parallel, following contract-driven development where each component is developed according to agreements with other components. It allows each responsible party to develop its components at its own pace without causing delays for the others. To test the components, mock-ups can be created for all components (best supplied by the team developing each respective component) so testing can happen during every Sprint. As the interfaces change over time, ideally in agreement with all the parties, the dependent components adapt themselves. New mock-ups can be developed to support testing.

Rather than fully developing base components first and then developing the dependent components, thus delaying release, Agile and contract-driven development allow parallel development of components by different parties. In fact, frequent integration testing allows early detection of problems and helps with resolution in a timely manner.

Test-driven development and automated unit testing should be used to ensure wide test coverage and the ability to run the tests frequently in different environments. Mock-ups can be used for component-level testing. Real integration tests then can be scheduled as and when actual components are available. The strategy for testing with mock-ups and real integration testing can be devised based on the delivery capability of each party and the release timelines.

Automated Testing

The testing of multiscreen solutions includes two phases. The first covers testing of the video plane where the transformation and delivery of video assets is assessed within quality-of-service (QoS) and quality-of-experience (QoE) parameters. The other testing phase concerns the control plane, where interaction among components and subsystems is tested to ensure desired implementation of various use cases.

Due to its very nature, testing the video plane is usually automated by the use of various tools. Automation in the testing of the control plane is often overlooked. As we have seen, these solutions are complex and are typically developed in an iterative fashion. And they contain various subsystems. The testing of a multiscreen solution thus involves repeated testing of individual components, subsystems and end-to-end scenarios in different environments. Hence, it pays to automate testing of the control plane as it delivers value throughout the solution lifecycle.

Test Automation of the Control Plane

Testing of the control plane typically involves each component in isolation (with all other components simulated), then progressively replacing each mock component with the real one. The same tests can be repeated. This approach is repeated throughout the lifecycle as appropriate and is tailored for unit testing during development, integration testing and performance testing for every release cycle. JUnit/NUnit tools and techniques can address the basic need for automation of these test cases, and they go well with the Agile methodology.

For more advanced automation, an automation test framework (ATF) can be developed. An ATF can set the environment, perform the tests, roll back the environment into its original configuration, generate test reports and update the status in external test management systems. An ATF would typically have a core engine for executing test cases but it abstracts and simplifies the development of the test cases with a ready-made request/response template along with support for multiple protocols that are commonly used in the control plane of a video solution (i.e., HTTP, RTSP, Web Services, REST, etc.).

With an ATF, the time and effort required to test components, subsystems and a complete solution in different environments can be drastically reduced. This framework can be easily extended and enhanced to accommodate growing needs and emerging requirements. The cost of developing such a framework is easily offset if the multiscreen solution has an enduring lifecycle.

Test Automation of the Video Plane

Testing of the video plane fundamentally addresses two specific quality requirements: QoS and QoE.

QoS is the monitoring of the discrete infrastructure components such as servers, streamers or network traffic (IP packets, etc.) that reflect the basic performance as seen by the operations team. QoS metrics are generally device- or transport-oriented, such as CPU usage, memory usage, packet losses, delay or jitter.

QoE metrics such as the time to search the content, switch to different content, etc. are important for customer satisfaction/retention and your brand reputation since they directly measure the quality of your service as experienced by the consumers.

QoE is a measure of a user's experience with the multiscreen solution. QoE metrics are user-centric – e.g., the time taken to return search results while looking for an asset, the time taken to start streaming once the content is requested, etc.

Both QoE and QoS are necessary to deliver a comprehensive monitoring and testing system. QoS monitoring is well suited for troubleshooting and root-cause analysis whereas QoE provides metrics that the customer will directly perceive as a quality parameter. To troubleshoot issues reported by QoE metrics, QoS metrics are useful.

Many leading QoS vendor monitoring tools such as JDSU, Ineoquest, Tektronix, etc. are used by operations teams to protect their investment (i.e., Cap-Ex) in video delivery infrastructure and reduce their Op-Ex by reducing the time to troubleshoot and fix problems.

While the benefits of QoS monitoring are well understood within many organizations, the advantages of QoE monitoring are often overlooked. QoE metrics such as the time to search the content, switch to different content, etc. are important for customer satisfaction/retention and your brand reputation since they directly measure the quality of your service as experienced by the consumers. QoE is especially important in the multiscreen world, since organizations typically control the video delivery networks but not the devices on which their services are consumed.

Managing video quality in unmanaged devices is a challenge, and having a sound test automation strategy for QoE is imperative. Organizations need a testing solution that automates users' behaviors on any device and records the response to simulate actual user experience. Without an appropriate automation strategy for testing video on multiple devices, the reliance on manual testing can introduce errors and delay the release of a service into the market. Vendors such as Witbe have great tools that allow you to emulate user interactions.

Looking Forward

While multiscreen video is an established capability, adoption is projected to only increase. Consumers will come to expect it from all content providers and operators as a de facto offering and not see it as a premium option. Rather than getting bogged down by the multitude of challenges faced by engineering, providers and operators would do well to engage with SI partners that can not only work around these challenges but can accelerate deployment of multiscreen offerings by leveraging their technological capabilities and resources. An effective SI partner can play a significant role in making the multiscreen vision a reality.

As such, we recommend a partner with the following attributes:

- The perfect blend of consulting skills to understand and enhance your solution vision and technical skills to lead its implementation.
- Vast experience in leveraging third-party solutions, with the ability to strongly recommend suitable options to meet your organization's needs and help make build-vs.-buy decisions.
- Well-established processes and tools adoption for Agile development and automated testing to rapidly implement the solution with best-in-class quality.
- Mature program management capabilities and delivery processes to run the entire development and deployment cycle in an efficient and cost-effective manner.

Delivering multiscreen solutions may be daunting but the right SI partner can help you deliver the innovation at the speed today's consumers have come to expect.

About the Author

Mehul Shah is an Associate Director within Cognizant's Engineering and Manufacturing Solutions (EMS) Practice. He has 18-plus years of professional experience, with a focus on product engineering and systems integration. Mehul has led large-scale programs for IP-based video delivery platforms that involve third-party product implementation as well as custom development. He has successfully helped various companies in the communication and media industry deliver multiscreen video solutions. Mehul can be reached at Mehul.Shah2@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 75 delivery centres worldwide and approximately 199,700 employees as of September 30, 2014, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.

World Headquarters
500 Frank W. Burr Blvd.
Teaneck, NJ 07666 USA
Phone: +1 201 801 0233
Fax: +1 201 801 0243
Toll Free: +1 888 937 3277
Email: inquiry@cognizant.com

European Headquarters
1 Kingdom Street
Paddington Central
London W2 6BD
Phone: +44 (0) 20 7297 7600
Fax: +44 (0) 20 7121 0102
Email: infouk@cognizant.com

India Operations Headquarters
#5/535, Old Mahabalipuram Road
Okkiyam Pettai, Thoraipakkam
Chennai, 600 096 India
Phone: +91 (0) 44 4209 6000
Fax: +91 (0) 44 4209 6060
Email: inquiryindia@cognizant.com