

Benefits of SCORM

Interoperability

According to Rustici Software (2010), the primary benefit of SCORM is interoperability because most eLearning content is integrated into an LMS. Similarly, if you produce an LMS it is certain that your clients will want to import content from a number of different sources into that LMS. SCORM allows this integration to happen seamlessly and simply. Without SCORM, integrating with other vendors is a time-consuming and costly process. SCORM will make your organization more efficient and reduce your support costs.

Interoperability is the ability to take instructional components developed in one location with one set of tools or platform and use them in another location with a different set of tools or platform.

SCORM is the de facto industry standard for interoperability. If you ever hope to integrate e-learning content with an LMS, it will simply be expected that you are SCORM conformant. SCORM conformance is a prerequisite to selling products in the e-learning space.

E-learning content that can only play in one LMS can only be sold to a very limited audience. Similarly, an LMS that can only play content designed for that LMS does not have wide appeal. Content that can be played *anywhere* and LMSs that can deliver *any* content are far more useful and have dramatically more appeal to customers. There is a huge ecosystem of e-learning products that all work together because of SCORM. Joining this marketplace significantly increases your market potential.

Reusability

ADL created an entire market ecosystem for e-learning by trying to save the government money by enabling content to be reused across the different parts of the DoD and other government agencies. Reusability is the flexibility to incorporate instructional components in multiple applications and contexts.

The biggest selling point for SCORM was actually reusability. Interoperability is very important, and it was the ultimate market driver for acceptance of SCORM, but reusability received all the glory early on. Interoperability has been a huge success for SCORM. Reusability has had decent success, but because it was over-hyped, it is often seen as a disappointment and leads some to call SCORM a failure.

To examine reusability in SCORM, it is helpful to take a step back and look at the long term vision ADL has for SCORM. In an ideal world, online training will be customized for every learner. Every individual has a different background, different competencies, different learning styles and different needs. Training should take these differences into account and adapt to the learner's specific needs. Ideally, it should be the LMS that handles this adoption, automatically. The system should be aware of the user and should have access to a large repository of "training components" that it can choose from to provide the best experience to this learner.

SCORM specifically addresses the ability to create this large repository of training components. For an automated system to be able to aggregate a customer tailored course, these training components need to be small discrete chunks, they need to behave in a standard manner, they need to be describable and they need to be assembled in a sequence. These technical needs are precisely what SCORM enables. The training components are called SCOs. They have a common packaging scheme, a common run-time behavior, they are described by metadata and they are put together in sequenced packages.

This long term vision is still on the horizon, it is something the industry is striving for but has not yet achieved. SCORM is a necessary first step down this path, but it alone is far from sufficient to achieve the grand vision. SCORM is a technical enabler, but there are still many barriers to overcome. Some barriers are technical but many barriers are not.

In organizations that produce or consume a lot of content, there are many opportunities for reusing modular content. SCORM enables SCOs (chunks of training) to be easily shared and reused across different courses. As an example, imagine a car company that trains all of its sales representatives on the features of each car model. It's likely that there is a standard sound system that is shared across several car models. Producing training in discrete chunks would allow this company to share the sound system SCO across the training courses for each car model. This form of reuse reduces the amount of training that needs to be developed and simplifies the updates that need to be made as the training changes. SCORM provides the technical infrastructure to enable this kind of reuse and for large training organizations it is an extremely valuable cost savings.

There are non-technical challenges that have prevented this form of reuse from becoming widely adopted. Challenges include:

- Creating an efficient repository to store and identify reusable content
- Creating effective modular training that is independent of context requires thoughtful instructional design
- Overcoming intellectual property restrictions
- Tracking the deployment and use of SCOs to ensure that they are properly maintained

These obstacles can all easily be overcome within an organization, but they present a large obstacle to creating a SCO-based training marketplace (IP restrictions are the big barrier currently). The fact that this SCO-based public marketplace doesn't yet exist causes some to call SCORM a failure. The naysayers point to this fact and say that SCORM doesn't enable reuse and that all the hype is overblown. I disagree with their assessment; SCORM has done everything it can from a technical perspective. The interoperability it provides is, taken alone, enough to call it a success—even without the much-hyped inter-organization reuse.

Accessibility

This is the ability to locate and access instructional components from one remote location and deliver them to other locations.

SCORM certainly does enable accessibility. SCORM courses are all packaged in a way that makes them easily portable across systems. A SCORM course can be delivered from any location without reconfiguration or complex installation. SCORM provides the technical framework for remote content to be cataloged and discovered.

SCORM does have one significant barrier to accessibility. The cross domain scripting problem prevents SCORM content from one server from being played on another server without physically moving the content (or jumping through some complicated hoops). This limitation is less than ideal.

Adaptability

This is the ability to tailor instruction to individual and organizational needs.

SCORM enables adaptability in two ways. It allows content authors to manually mix and match SCOs to create unique training programs for different groups. It also allows content authors to write sequencing rules that adapt the instructional content as the learner progresses based on the learner's input and mastery. Both of these mechanisms have enjoyed mixed success. The manual reassembly of SCOs is very possible from a technical sense, but has suffered from the non-technical limitations discussed under reusability. Instructional content that adapts itself at run-time currently relies on the SCORM sequencing specification. This specification works well for some scenarios, but not for others. The sequencing specification is widely considered to be the weakest part of SCORM.

Affordability

This is the ability to increase efficiency and productivity by reducing the time and costs involved in delivering instruction.

SCORM has certainly reduced the costs of integrating content into a LMS. There are some who still say that implementing SCORM itself is too hard and too expensive, largely due to the sequencing specification. Sequencing *is* quite complex for LMS vendors to implement and can be overkill for content developers to implement in situations where it isn't necessary. When applied appropriately, however, SCORM generally results in great cost savings.

Durability

This is the ability to withstand technology evolution and changes without costly redesign, reconfiguration or recoding.

SCORM has done pretty well with durability, but then again, isn't the entire role of a standard to keep things constant? SCORM has done a good job of using established, mainstream technologies that will be widely supported and available for a long time.