



## **Industry Best Practices For eLearning Development**

Q1 - 2007

By  
Joseph Gorup  
Catavo, LLC

All content Copyright 2007, Catavo, LLC

## TABLE OF CONTENTS

Executive Summary .....	3
Why are Best Practices Important to eLearning Development? .....	3
What next?.....	4
eLearning Development.....	5
Mass-author or “the funnel”.....	5
Collaborative Tools .....	7
eLearning Deployment .....	8
eLearning Best Practices .....	9
Content Development Best Practice Considerations .....	12
Workflow and Content Development .....	12
Categories of eLearning.....	14
Media Management .....	14
Source Code .....	15
Technical Considerations.....	15
Conclusion .....	17

## Executive Summary

### ***Why are Best Practices Important to eLearning Development?***

The creation of eLearning is a much more complicated effort than most people understand. Looking at the end product of many eLearning development projects, the “course” may look fairly simple. A table of contents, back and forth buttons, a few questions and a summary page, and various media elements (e.g. audio voice-over, drag-and-drop interaction, etc.) are common elements of the vast majority of content on the market.

Why is creating a seemingly simple end-product so complicated? The following describes basic challenges involved in both the development and deployment of eLearning.

#### **1) Development**

*The creation of eLearning requires input and coordination of multiple people.*

eLearning development requires a collaborative effort between a number of different roles. A typical “team” involves an instructional designer who organizes the course, one or more subject matter experts, one or more media development specialists (e.g. graphics or interaction developers, audio, video, technicians), a copy editor, and any number of people to review and approve the final deliverable.

Unfortunately, organizing this team is often done through a myriad of emails, conference calls, and meetings. At best, this process is somewhat inefficient. At worst, the final deliverable misses the mark or never gets completed.

#### **2) Deployment**

*The delivery and execution tracking of eLearning involves challenging and very technical communication programming.*

The assumption is that one wants to track the delivery of eLearning to a learner population. In other words, as opposed to simply posting a course on a website, one desires to have a learner’s progress, score and/or completion of a course tracked in an LMS.

In short, the means by which a course “talks” back to the LMS is a very complicated business. There are technical guidelines to help ease this burden. However, these guidelines evolve, are somewhat open to interpretation, and overall are just very technical.

Why is this communication process so challenging? Consider this example: If 5,000 learners wish to take a course – that means 5,000 individual conversations must occur between each learner’s desktop browser and the LMS. This poses a number of challenges such as browser support, security considerations, and general usability challenges.

Compounding these problems is that organizationally, training and development personnel typically do not have the background or tools required to address the challenges noted above.

---

While adept in communicating and teaching, a background of organizing a multi-disciplined team to execute on a very technical development effort is not often found in the profile of those often chartered with creating eLearning.

***What next?***

The above outlines basic challenges involved in eLearning development, and the following expands on these challenges and provides a set of practices that can be implemented to mitigate the risks and lead toward successful development, management and delivery of eLearning.

## eLearning Development

A fundamental business problem is that organizations cannot efficiently create custom courseware. After spending large sums of money on learning management systems or in other education infrastructure areas, many organizations are doomed to managing instructor-led courses or only delivering eLearning programs based on off the shelf courseware. *As a result, the desired goal of capturing and delivering corporate-specific know-how through eLearning is lost.*

### **Mass-author or “the funnel”**

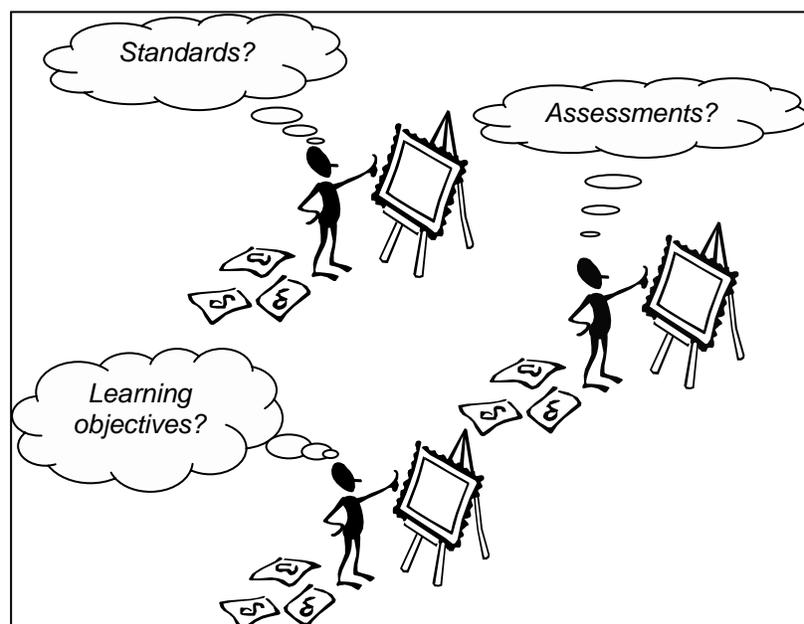
*Neither works on a large scale*

Many attempts to create courseware have resulted in inefficient and un-scalable business processes. The processes that have evolved are complete opposites. One side is a mass-author model; the other a single author or “funneling” approach. In the mass-author model, organizations turn all employees into “authors” in an attempt to capture their knowledge. In the funneling model, everyone involved in creating courseware funnels their knowledge into one person trained to create the formal courseware. Ironically, most often this “one person” is not from the training and education organization, but rather from the Information Technology staff.

These unworkable business processes are largely a result of organizations attempting to structure themselves around an immature “authoring tool” marketplace. More specifically, there is a schism between these low-end “authoring tools” and high-end “all encompassing” learning content management systems (LCMS). Neither end of the spectrum adequately addresses the basic business problem and have led to inefficient authoring models of executing development efforts.

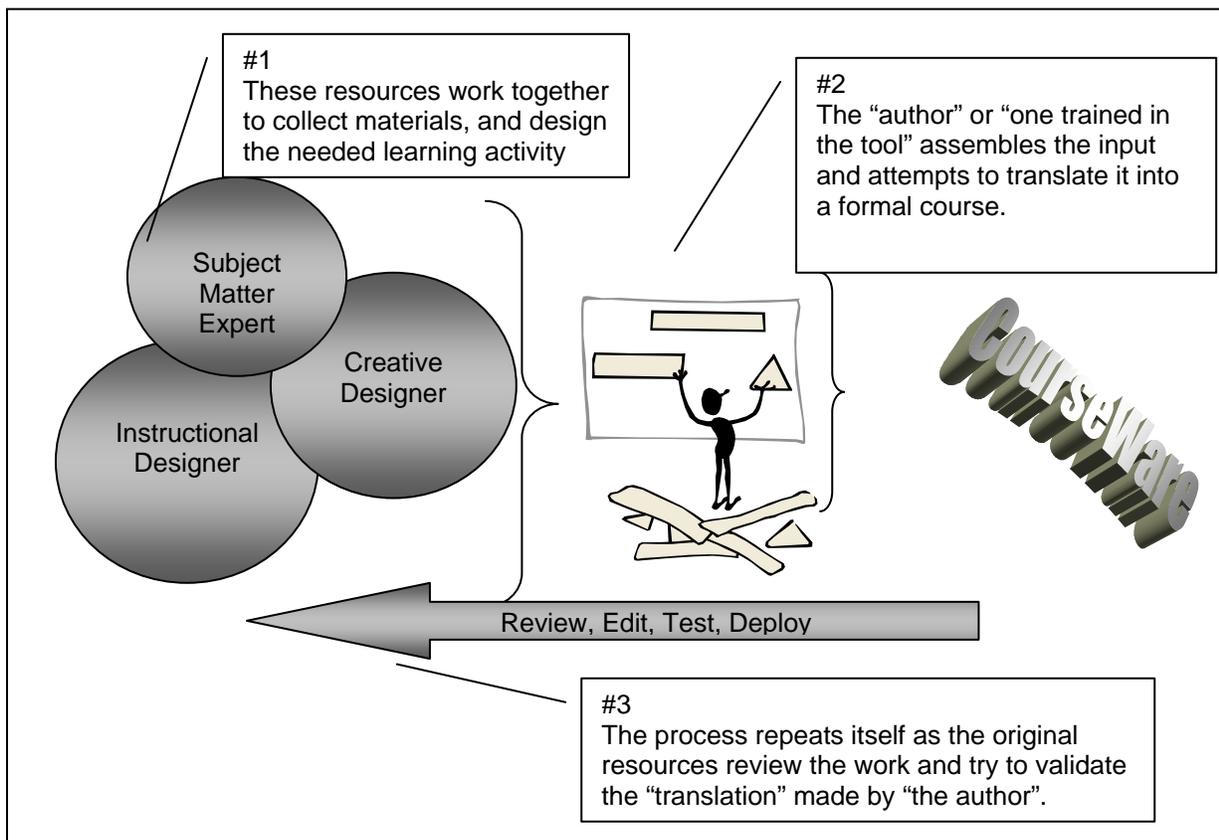
### **The Two Processes**

The mass-author model explains the numerous products focused on taking traditional end user tools (e.g. MS Word, PowerPoint) and converting their output into “courses”. While this can be an effective means of gathering raw content, it does not address areas such as instructional design, use of assessments, appropriate use of multi-media, or general training standards. These issues are in addition to the fact that “authoring courseware” is not in the job description of everyone in the organization.



As the above diagram shows, by turning everyone into an author, *every* person struggles with *all* of the aspects of courseware development. The thought of maintaining standards or any notion of re-use is abandoned. This is obviously not an efficient or scalable model.

With the other model, all courseware materials are funneled to the select person (or outsourced firm) capable of using the content development tool. Often this select person is someone from the IT staff with a programming background. In this model, all the roles involved in the effort submit their pieces to the “author” who creates the actual course. The pieces are communicated to the author in a variety of ways from meeting notes, to documents, to PowerPoint “mock ups”. The following image provides a visual on this “funneling” process.



This obviously inefficient process gets worse when one factors in individual work schedules of those involved. Simply trying to coordinate the team members’ schedules is a complete job within itself, almost requiring a “courseware process manager”. Materials must be gathered and course versions reviewed (alpha, beta, etc.). Changes need to be communicated to the developer; a wait ensues while the changes are implemented. Another delay occurs while final confirmation of the product is given. In some cases this process is longer than the shelf life of the course being developed. The course content may likely need updating before it is ever published.

Some organizations use a combination of the above methods. For example, initial content may be generated via PowerPoint/Word, structured by a broader team, and then finally “authored” by someone using the more sophisticated tools. While creative, this process introduces further

complications. For instance, once the content is integrated into a formal course, how is it maintained? In PowerPoint? In the worst case, this hybrid approach ends up maximizing the inefficiencies of both processes.

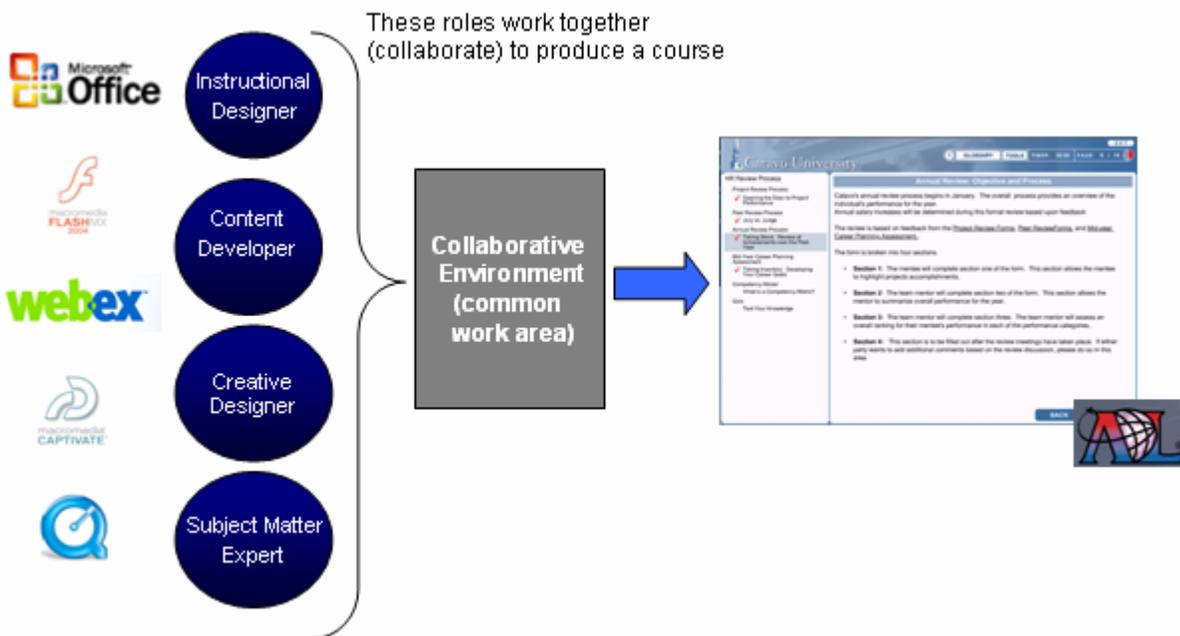
## Collaborative Tools

*Emerging development tools to meet business needs*

In order to meet business needs, collaborative eLearning development tools are emerging. *This market is characterized by tools aligned with the courseware development process rather than dictating a mass-author or "sole-author" models.* In other words, this market takes a "process-centric" view versus an author or "learning model" centric view. By embracing the courseware development process, technology can automate and add value as opposed to forcing organizations into something they are not. The following diagram illustrates where collaborative tools fit in the overall courseware development market.



Courseware development requires input from many disciplines. For example, instructional designers will establish learning objectives, while subject matter experts provide the teachable know-how, and creative designers contribute the enhancing media. Each role plays a part. To facilitate this process, each contributor must directly participate in the process. By providing a common work area or collaborative environment, people can work together.



## eLearning Deployment

The deployment of web-based courseware to a learner population is a technically challenging complex effort. Upon “launching” a course, various learner activities are communicated to the LMS. For example, as the learner progresses through the course, “bookmark” and “time in the course” information is sent back to the LMS to store.

There are two key “standards” that govern how a course should communicate with the LMS. These are the AICC and SCORM protocols. To understand the organizations that created these standards and the specifications behind them, see [www.catavo.com/documents/elearning-standards.pdf](http://www.catavo.com/documents/elearning-standards.pdf).

The phrase “standard” is put in quotes because there is a misconception about the state of these standards. While the standards these organizations define are, in fact, labeled as **standards**, they generally start out as a specification, which is a detailed, exact statement of the functional requirements and particulars for something to be built, installed, or manufactured. For a specification to reach the status of an accredited standard, it must receive some sort of stamp of approval from an accrediting body. For simplicity sake, throughout this document we will refer to the “standards” however, *one is forewarned of there are wide variations in how these specifications are implemented from vendor to vendor.*

Here is one example of how interpretation of the specification manifests itself. While the following example may seem trivial, dozens of hours were spent in trying to solve this issue:

*The customer had one LMS for internal users and another for its distributors and suppliers. One course worked flawlessly in the internal LMS but seemingly would not communicate with the other (e.g. bookmarks were not set, scores not reported, etc.). Hours were spent looking at web servers, client configurations, log files, etc. After much detailed analysis, it was learned that one LMS was expecting data from the course terminated with a “carriage return” (hex code “OD”) and the other was expecting both a carriage return AND a line feed (hex codes OD and OA). As one may expect, debugging communication data down to the hexadecimal level requires significant technical know-how.*

The goal of eLearning standards is to provide fixed data structures and communication protocols for eLearning objects and cross-system workflows. This enables interoperability between applications, such as an LMS and third-party or in-house developed content, by providing uniform communication guidelines that can be used throughout the design, development, and deployment of learning objects. When these standards are incorporated into off-the-shelf products, developers can base their purchasing decisions on quality and appropriateness rather than compatibility.

Unfortunately, the typical components within a learning environment are supported by multiple products and vendors. Not surprising, if all the points of interoperability among eLearning components vary from vendor to vendor, then it is very difficult and costly to implement an integrated learning environment.

In the end, however, it is crucial to understand that corporations and institutions that seek to make eLearning offerings available to a learner population generally underestimate the technical nature of eLearning. Typically, the focus is on the “tool” aspect of course building. Instructional design, graphical elements, and user interaction concerns all play a part in the “front end” aspects of building compelling eLearning. However, these concerns usually come at the expense of very important “back end” issues, such as interoperability, communication standards, and deployment constraints. These issues should be treated with as much attention, if not more, than the traditional ones have.

## eLearning Best Practices

1. **Define a “road map” for eLearning development efforts.** A road map is necessary to ensure that team roles and responsibilities are clearly defined. Also, a defined plan is much easier to manager. When a road map is not clearly defined, miscommunication between team members and a misunderstanding regarding what needs to be done is more likely to take place. (See *“eLearning Development Roadmap”* – p. 13)
2. **Provide an eLearning development platform/toolset as a baseline for content creation and management.** A common development platform allows all involved to adhere to a framework of standards that is easily repeatable and produces consistent output. Failure to provide a starting point or default platform for development leads to a potpourri of challenges when upgrading, managing skillsets, etc. (See *“Collaborative Tools”* – p. 7)
3. **Provide a collaborative environment where teams of people work together.** eLearning development requires input from multiple people (instructional designers, graphics, subject matter experts, etc.) and providing a place for teams of people to work together is necessary. Too often, a common “workspace” results in disjointed pockets of development with organizational content buried in email inboxes and “My documents” directories. As a result, many companies evolve into either a “funnel” or “mass author” approach to creation - both of which lead to inefficiencies. (See *“Collaborative Tools”* – p. 7)
4. **Provide a means of sharing assets across an enterprise. However, one should focus on “use” before looking for “re-use”.** Efficient storage and tracking of learning assets across the organization leverages the investment in content creation. Lost or misplaced learning assets are a waste of time and money, and many times cause undue rework by team members. (See *“Media Management”* – p. 14)
5. **Understand there are categories of eLearning.** These are termed Level 1 (simple informational courseware) through Levels 2 and 3 (basic instructional courses) to Level 4 (immersed simulation courseware). It is important to remember that different tools, techniques, and budgets apply across this range. It is critical to understand the complexity of the learning required so that the appropriate tool(s) can be utilized effectively. Using the wrong development tool can seriously handicap the type of learning that needs to be created, often crippling the learning experience based on an incompatibility between form and function. (See *“Categories of eLearning”* – p. 14)

6. **Provide instructional design templates and examples for each category of eLearning.** To enable the development of rapid eLearning, a repeatable framework allows resources to focus on content, not structure. Without templates and examples, time is wasted as non-developers struggle to create course structures that are consistent. *Learn more about content management at [www.catavo.com/documents/elearning-content-management.pdf](http://www.catavo.com/documents/elearning-content-management.pdf).*
7. **Understand that eLearning components have “source code”.** If you lose the source code, you cannot update the course. Adequate source code control protects the programming investment in your course and helps future resources more easily maintain it. Failure to version and archive source code can create an un-maintainable mess when the course needs to be updated. (See “Source Code” – p. 15)
8. **Consider the “branding” of the courseware in its design and deployment.** It is advantageous to separate content from presentation in order to provide the ability to customize the appearance of courseware for different audiences. If this is not followed, either stale player skins or the necessity to update multiples copies of the courseware player can result.
9. **For those submitting content outside of the “provided platform,” provide technical examples of AICC and/or SCORM based courseware that has been tested to work in your environment.** Highly complex communication standards necessitate up-front planning and preparation to ensure that courses can “talk” to the LMS without any issues. Relying on crossed fingers when third-party courseware is integrated into a learning environment will almost certainly fail due to an unforeseen technical problem. (See “Technical Considerations – Communication Standards” – p. 15)
10. **In general, restrict the notion of “custom courseware” as well as the toolset people use to provide content.** A consistent development platform avoids the need to use third-party development tools and provides a technology base that is easier to implement from. When this is not followed, custom courseware can cause confusion among development resources and introduce technical interoperability issues post-deployment.
11. **Set expectations that standards such as AICC and SCORM are (a) loose standards, and (b) very technical in nature.** Organizations need to ensure they have the resources to decipher and implement complex technical specifications. Assuming that adherence to industry standards is consistent across LMS’s and course development tools is a recipe for disaster. (See “eLearning Deployment” – p. 8)
12. **Test, analyze and size the deployment environment (e.g. your corporate network) before deploying eLearning.** Use the output of this exercise as feedback to the development guidelines. You must get the proper resources with an understanding of the enterprise’s network involved to gain the proper communication understanding. An absent deployment strategy will almost certainly cause issues after the course development stage. (See “Technical Considerations - Deployment Constraints” – p. 16)

13. **Pending the LMS choice, be prepared to support AICC and multiple versions of SCORM.** Learning management systems differ in their support of so-called industry standards. If multiple standards are not considered, the courses you purchase or custom develop may have communication issues with the LMS, depending on the communication protocol chosen. (See *“Technical Considerations - Communication Standards”* – p. 15)
14. **Understand that mis-application of even one aspect of the communication between a course and the LMS can wreck havoc with the learner population.** Sometimes, even the smallest of implementation problems can cause huge problems across the installed base. The possibility exists for incredibly hard issues to debug. For example, consider a situation where 2000 people in an organization take an exam at the end of a required course. 900 of those people scored 98% for the course, but were marked incomplete in the LMS. The cause turned out to be an incorrectly programmed algorithm that caused this result because all of these learners answered “c” for question number nine in the exam. Problems like this can sometimes cause huge headaches while tracking down.
15. **Multi-language options should be considered up-front.** Global companies require global training. If the tools used to develop content do not support multi-language (i.e. Unicode for Chinese), it would then become very costly to convert content to another language. If this option is not considered up-front, you may need to start from scratch.

## Content Development Best Practice Considerations

### *Workflow and Content Development*

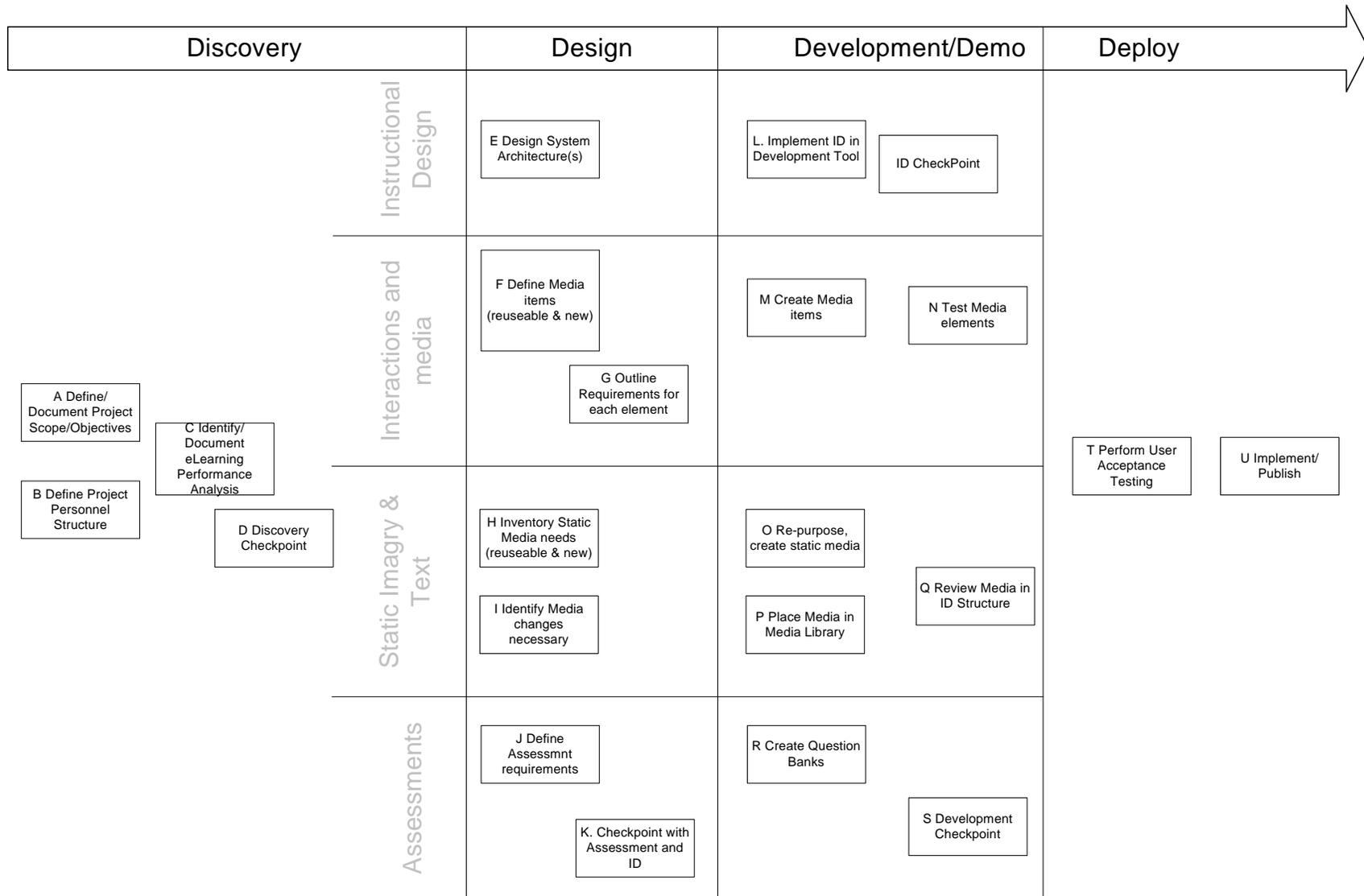
The process to create and publish training content to the web includes several phases, which may require different tools. The actual content must be outlined and developed by Instructional System Designers (ISD) and Subject Matter Experts (SME). Multimedia content such as graphics, photos, PC screen captures, audio files, video clips, and reference documents must be collected. At this point, a tool to develop the presentation the learners will experience is required. The content is then placed on an appropriate content server and the LMS is made aware of the content in order to make it available to the learner.

The tools for creating, capturing and preparing the multimedia are usually different from the training content development tool. These include packages such as Adobe Photoshop, Adobe Premiere, Sound Forge, Adobe Acrobat Pro, Macromedia Flash, Adobe Illustrator, etc. Given the number of tools that are available, one should focus on meeting content standards rather than dictating the development tool.

Traditionally, eLearning content development has been a linear process. Subject Matter Experts and Instructional Designers would produce the content and hand it over to a resource serving an artificial role of “author.” The author would then combine the text, graphics, multimedia and source code to create a finished course. The author then becomes a bottleneck through which all content, media and programming must flow through in order for the course to be completed.

Recently, we have seen the emergence of a few web-based tools which eliminate the need for an author and allow each development resource to collaborate online, simultaneously, and create eLearning courses just as compelling as before. To illustrate this new development approach, a four-phased “eLearning Development Roadmap” is illustrated on the following page. In it, there are clear demarcations of responsibility by role, dependent on the task. Given the right tool, all the tasks needed within the “Design” and “Development/Demo” columns can be done at the same time by different resources scattered around the globe. This is a powerful new paradigm in course creation that didn’t exist only a few years ago.

## eLearning Development Roadmap



## **Categories of eLearning**

Web-based training content and training content in general can be divided into categories ranging from very straightforward presentations similar to an information briefing, to very complex such as training for an involved procedure utilizing advanced equipment. A flight procedure required of a pilot in a modern jet aircraft would be an example of complex training content. The Computer Based Instruction industry uses categories named levels to describe the complexity of content. Level 1 content is straight forward with linear progression from frame to frame or screen to screen. Level 4 is the highest category using this scale. Level 4 includes complex presentations to the student and complex progression through the lessons. For example, pilot training requiring a simulation of the aircraft cockpit is an example of Level 4 content.

Traditionally, different tools have been appropriate for use when creating Level 1 and some Level 2 web-based training than are required for development for other Level 2, 3 or 4 courses. Level 1 and 2 content development using a programming environment required for level 4 development can be accomplished but the level of effort will be higher by at least a factor of four and the employees required must have highly specialized experience as a programmer. A Level 1–2 tool usually cannot develop most higher levels of training. Again, recent developments in the tool industry have seen the creation of a few web-based tools that will allow the creation of Level 1, Level 2, and some Level 3 content development. Other Level 3 and Level 4 requirements can be augmented through other multimedia tools, which can then be integrated into these same few web-based tools. Once again, this is a powerful new way of developing courses that didn't exist a few years ago.

Testing of learners is another important capability of web-based training delivery systems. The same web-based content development tools allow resources to develop tests and activities to quiz students within the lessons. Exam results are then communicated to the LMS via the appropriate communication protocols.

It is a recommended standard that course content should be developed to function properly on displays configured at 800X600 pixels. All content considered training (not just resources) must keep the learner's position (bookmarks) in the content and completion status and/or grade. The AICC method is preferred as it has proven most reliable in many environments (see *"Communications Standards on the next page"*).

## **Media Management**

Before web-based development tools were created, course authors were tasked with having to locate media assets required for a course. Unfortunately for the author, media could be on another colleague's machine in the same office or overseas, or on an unexposed server somewhere in another office. Since the advent of web-based tools, we have seen an introduction of the media library which allows online, collaborative management of media assets within a single, virtual environment. By using a media library, organizations can allow resources all over the globe to contribute and access media assets within an organized structure. Within the media library, all categories of media asset can be stored and tracked, including:

- Audio (music, narration)
- Documents (PDF, scanned, Word, text)
- Images (clip art, photos, graphics)
- Multimedia (animation, simulation, video)

## **Source Code**

In the past, typical eLearning courses have contained source code that was difficult to maintain/update should a coding resource or freelancer not be readily available after development was completed and the course rolled out. A development framework that allows non-programmers to assemble the same quality of eLearning breaks this requirement of having to rely on niche technical skills that may be difficult to re-acquire.

Beyond simple resource acquisition, however, is the issue of source code control and code archiving. If a development tool requiring any type of programming code is used but is simple enough for non-programmers to use, mechanisms need to be put in place to ensure that the source code is maintained by a version control system and older versions of the code archived appropriately. These mechanisms are not encompassed within the development tool itself, so it is crucial that someone with technical knowledge set up the relevant systems and have them be regularly used to protect your investment in the course functionality.

## **Technical Considerations**

There are three areas that relate to the back-end technical aspects of course delivery and how the course will communicate with an LMS. These can be categorized into the following areas:

- Communication Standards
- Deployment Constraints
- Interoperability

### **Communication Standards**

The communication interface is how learning resources exchange information dynamically, such as while a student is taking a lesson. Currently, both the AICC and the ADL specify communication interfaces, while IEEE and IMS specifications do not yet explicitly identify a communication interface. *Learn more about these organizations at [www.catavo.com/documents/elearning-standards.pdf](http://www.catavo.com/documents/elearning-standards.pdf).*

For content developers, the most widely accepted communication protocol is Hypertext AICC Communication Protocol (AICC HACP). The AICC CMI guidelines describe a common set of data records about student performance and lesson history that are useful for tracking student performance across sessions. The AICC also manages an independent test lab that certifies compliance with the AICC LAN or HACP protocols for either content or management systems.

The trend among LMS vendors is toward an alternate AICC communication protocol that is also shared with the ADL, called "API Communication," and often referred to informally as the LMS API. This API is the run-time communication specified in the ADL SCORM document, and acceptance and implementation are accelerating rapidly.

While these guidelines exist, it is important to understand that they are highly complex, due to how communication from a course to an LMS must occur. A multitude of communication issues due to patch levels, server configurations, firewalls, and networking protocols all contribute to an environment where issues can arise, so it is imperative that the proper planning and preparation take place to anticipate these issues in advance.

### *Deployment Constraints*

From a network-level standpoint, deployment issues typically arise due to a lack of practical experience and understanding of an overall deployment strategy. It is crucial to involve resources with a high-level understanding of the organization's network in order to map out the data communications paths along the network from a user's desktop to all relevant servers (content, media and learning management). Too often this step is skipped until after courses are built, at which point any type of planning effort is certain to encounter issues. Some common pitfalls of a poor or absent deployment strategy are:

- Course is too large – content developers create a course many megabytes in size, but the network is incapable of delivering such a large course to the desktop within an acceptable amount of time, or the network cannot handle such large file sizes.
- Media format issues – course creators include courses with certain media types (e.g., Quicktime movie files), and the user population does not have the proper media player installed on their desktop.

It is recommended that to prevent such issues, a test environment be setup to test all deployment concerns ahead of time.

### *Interoperability*

Related to deployment constraints, there are a number of desktop-level interoperability issues which must be addressed prior to rollout of eLearning across the organization. These issues can (potentially) affect every user differently, since the desktop environment is much more apt to be customized and altered by the user. Following are some examples of issues that can impact course delivery and performance at the desktop, but this is by no means an all-inclusive list:

- General browser compatibility
- Use of browser plug-ins
- Poor understanding of user's desktop configuration

---

## Conclusion

Content standards for eLearning are an essential element of creating compliant courses that meet the actual and anticipated needs of the organization and its learners. Because they might overwhelm, it is recommended that you focus on the capabilities they address – workflow and content development, content management and technical considerations. For some, standards may have fallen off the eLearning radar screen. But in reality, standards are and continue to be a part of all aspects of learning.

In addition, the broad base of vendor support for these standards will provide an added bonus for organizations because content will be more quickly and easily deployed to a larger number of customers, regardless of LMS environment. It is comforting to know that solutions exist to create a more efficient workflow while insulating resources from the often complicated details of eLearning standards and specifications.

Most important, however, is the fact that since eLearning standards are constantly evolving, it is crucial to continue educating yourself about developments among the standards organizations as well as gaining expertise about the technical issues at hand.



Catavo offers industry-leading eLearning consulting services with a proven methodology and implementation track record with leading Fortune 500 clients. Please contact us to learn how we can help you with your eLearning requirements.

[www.catavo.com](http://www.catavo.com) | [sales@catavo.com](mailto:sales@catavo.com) | 630-393-0022

All content Copyright 2007, Catavo, LLC