

Discussion Guide

Introduction to Content Management

A companion set of open-ended discussion questions for all eight lessons of the course — designed for classrooms, study groups, reading circles, and team workshops. There are no single right answers here; each prompt is meant to spark debate, reflection, and the application of ideas to real situations.

HOW TO USE THIS GUIDE

In a classroom: assign a lesson as reading, then work through 4–6 questions as a group. Questions build from recall toward judgment and application.

In a study group: let each member champion a different question and lead the conversation.

On a team: use the prompts tagged **APPLY** to connect the ideas directly to your own CMS, content, and processes.

8 lessons · 96 discussion questions · No prior experience required

What Is Content Management?

Focus: The history of the discipline, the definition of content, content vs. data, discipline vs. software, and what a CMS does and doesn't do.

- 1 The course claims content management is "older than the Web" and points to the Library of Alexandria. What do a modern CMS and an ancient library genuinely have in common — and where does the analogy break down? **ANALYZE**
- 2 Content is defined by two properties: creation through editorial process, and intention for human consumption. Can you think of something that has *one* property but not the other? Is it content? **DEBATE**
- 3 "Content is never 'right,' it's just 'right now.'" Do you agree that all content is perpetually provisional, or are there kinds of content that are genuinely "finished"? **DEBATE**
- 4 The distinction between content and data is said to be defined by the *services* we apply, not the information itself. Take one piece of information from your own life and argue it could be either. **APPLY**
- 5 Why does the course insist on separating the *discipline* of content management from any particular *software*? What is gained by holding that distinction firmly?
- 6 If there is "no Grand Unified Theory of Content Management," how should a newcomer decide which best practices to trust? **DEBATE**
- 7 Of the four things a CMS does — control, reuse, automation/assembly, editorial efficiency — which would matter most to a small nonprofit? To a global news organization? Why the difference? **APPLY**
- 8 The course lists four things a CMS does *not* do (create content, make marketing plans, format well, provide governance). Which of these failures do you think derails the most projects, and why?
- 9 Some clients called years after launch asking how to log in for the first time. What organizational failures lead to a brand-new CMS never being used?
- 10 In the homebuilding analogy, "you" are the prime mover, not the tools. Where in your own experience have people mistaken the tool for the outcome? **REFLECT**
- 11 "A CMS is a means to an end, never the end itself." How would you explain this idea to an executive who wants to buy a top-rated platform to "fix the website"? **APPLY**
- 12 Editorial morale is called "intangible but critical." How much should the *feelings* of the people using a system influence which system an organization chooses? **DEBATE**

Types of Content Management Systems

Focus: WCM, ECM, DAM, and records management; the blurry boundaries between them; the meaning of "enterprise"; and the concept of opinionated software.

- 1 The "big four" (WCM, ECM, DAM, RM) each excel at something different. If a single organization needed all four strengths, would you advise one blended system or four specialized ones? **APPLY**
- 2 DAM is said to be defined by what it *does* to assets rather than what it stores. Is a category defined by its processes more or less useful than one defined by its content? **DEBATE**
- 3 The lines between CMS types are "blurry," and systems get classified by market, community use cases, and features. Is this classification helpful to buyers, or does it mislead them?
- 4 "Enterprise" is described as a vague term that often just means "expensive." Where else in technology do you see vocabulary used more for positioning than precision? **REFLECT**
- 5 "All software is opinionated; the only question is how strong those opinions are." Do you find this liberating or troubling as a future buyer of software? **DEBATE**
- 6 Your affection for software is "directly proportional to how closely its opinions line up with yours." How can a buyer discover a system's true opinions *before* committing to it?
- 7 The "list of pages" example shows a set of built-in opinions that satisfies most customers. When is it wise to accept a system's defaults rather than customize? **APPLY**
- 8 "Requirements settling" — changing your use case to fit the software — is called more common than people admit. Is this pragmatic wisdom or quiet defeat? **DEBATE**
- 9 Some customers *want* software to enforce its opinions and treat a product as "hard-won consulting." When is outsourcing your methodology to a vendor a good idea?
- 10 Strongly-opinionated software "thrives in industry verticals" like museum collection management. What field that you know well would benefit from a strongly-opinionated tool? **APPLY**
- 11 The market trend is toward generalized platforms with "fewer opinions." Do you expect that trend to continue, or will AI push things back toward opinionated, do-it-for-you tools? **DEBATE**
- 12 Drupal (a WCM) gets used for enterprise content and Documentum (an ECM) for websites. What are the risks of using a tool far outside its intended category?

Content Modeling I — Types, Attributes & Objects

Focus: Reification, the anatomy of an attribute, attribute types vs. assignments, content types as wrappers, and the type-vs-object distinction.

- 1 Reification is "making the abstract concrete." Why is turning a fuzzy human idea into rigid fields one of the hardest parts of building any content system? **ANALYZE**
- 2 The street-address example shows structure trading flexibility for capability. Describe a piece of content where you'd deliberately choose *less* structure. **APPLY**
- 3 An attribute turns out to have many hidden parts (value, name, label, datatype, editorial element, validation). Why is it valuable for even non-technical editors to know this? **REFLECT**
- 4 Internal attribute names are often unchangeable once set. What discipline does that impose on the people first designing a model?
- 5 "A type contains no data." Why is it important to insist that all data lives in attributes and a type is merely a wrapper?
- 6 The distinction between an attribute *type* and an attribute *assignment* is subtle. Why does "Title on Article" need to be treated as different from "Title on Employee Profile"? **ANALYZE**
- 7 The cookie-cutter analogy separates the pattern (type) from the instance (object). Where else in software or life do you see the same pattern-vs-instance idea? **REFLECT**
- 8 The course jokingly proposes calling content types "Platonic Forms." Does philosophical framing help you understand modeling, or does it overcomplicate a practical task? **DEBATE**
- 9 Two types with identical attributes are "functionally the same type." When, if ever, is it worth duplicating a type purely for a different name (e.g. Article vs. Blog Post)? **DEBATE**
- 10 Modeling mistakes "reverberate for the entire life of your system." How should a team balance careful up-front modeling against the need to ship something?
- 11 Sketch a content type for a recipe. What attributes would you include, and which datatype would each use? **APPLY**
- 12 If you had to teach the type-vs-object distinction to someone in one sentence and one example, what would you say?

Content Modeling II — Relationships & the Shape of Content

Focus: Forms of relationships, referential integrity, the five shapes of content, matching a CMS to a shape, and the granularity-vs-usability trade-off.

- 1 "Content almost always exists in relation to other content." Can you think of any content that is genuinely standalone, with no meaningful relationships? **DEBATE**
- 2 Relationships range from explicit reference attributes to implied co-membership in a category. Which kinds are most fragile, and why do they break? **ANALYZE**
- 3 Why is referential integrity — not deleting something others depend on — harder to maintain as a content collection grows?
- 4 The five shapes are serial, hierarchical, tabular, network, and relational. Pick a website you use often and identify the shape of each major section. **APPLY**
- 5 WordPress excels at serial content but struggles with hierarchy; MediaWiki is the reverse. Why do tools become so specialized to a shape? **ANALYZE**
- 6 "Almost any system can be contorted to work with any shape" — but at the cost of heroic effort or a confusing editor experience. When is that cost worth paying? **DEBATE**
- 7 A single site usually mixes several shapes. What problems arise when one CMS must serve hierarchical, tabular, and serial content at once? **APPLY**
- 8 Josh Clark warns that heavy structure "forces editors to approach their content like machines." Where's the line between helpful structure and dehumanizing rigidity? **DEBATE**
- 9 The course warns that over-modeling is as real a failure as under-modeling. How would you recognize that a model has become too granular?
- 10 Requirements should drive granularity, "not hypothetical future ones." How do you resist the temptation to model for imagined future needs? **REFLECT**
- 11 Consider a restaurant's website. Which relationships between content objects (menus, dishes, locations, staff) would you model explicitly? **APPLY**
- 12 If your content's shape and your CMS's strengths disagree, is it better to switch tools or adapt your model? What factors decide?

Editorial Tools & Workflow

Focus: The content lifecycle, the CMS team, versioning, scheduling and expiration, workflow vs. approvals vs. collaboration, and permissions.

- 1 The lifecycle (create → edit → approve → publish → archive → delete) is "not a straight line." Trace a realistic path a piece of content might actually take, with loops and detours. **APPLY**
- 2 A single object can be published, in draft, and archived all at once. Why is this "layers of visibility" idea confusing for newcomers, and how would you explain it?
- 3 Five roles surround a CMS (editors, site planners, developers, administrators, stakeholders). Which role's needs do you think get neglected most often, and why? **DEBATE**
- 4 "'Editor' is a crude generalization." Why does treating all editors as one undifferentiated group cause problems?
- 5 Versioning enables rollback, audit, and comparison. Which of these matters most in a regulated industry like finance or healthcare? **APPLY**
- 6 Scheduling and expiration automate publishing and retirement. What could go wrong if a large site relied heavily on automatic expiration? **ANALYZE**
- 7 The course distinguishes workflow (movement) from permissions (authorization). Why is conflating the two a common and costly mistake?
- 8 Teams often ask for "workflow" when they really want "collaboration." How would you diagnose which one a team actually needs? **APPLY**
- 9 "Rigid approval chains can get in the way of the informal back-and-forth that makes content good." When is heavy process genuinely worth the friction? **DEBATE**
- 10 Permissions are granted through roles and groups rather than individually. What are the trade-offs of role-based access versus per-person control? **ANALYZE**
- 11 Dependency management surfaces "orphaned" content. Why does unused, unreferenced content accumulate, and whose job is it to prune it? **REFLECT**
- 12 Design a minimal approval process for a five-person team's blog. What is the fewest number of steps you can justify? **APPLY**

Content Delivery & Aggregation

Focus: Separating content from presentation; content, artifacts, and channels; templating and the surround; content aggregation; and the tyranny of the tree.

- 1 Separating content from presentation lets one source feed many channels. What is the hidden *cost* of that separation, in effort or complexity? **ANALYZE**
- 2 Content, artifact, and channel are carefully distinguished. Take a single article and describe three different artifacts it might become. **APPLY**
- 3 Because content is "write once, read many," heavy investment in delivery pays off. Where else does the WORM principle justify up-front effort? **REFLECT**
- 4 Templating languages are deliberately *less* powerful than full programming languages. Why would a system intentionally limit what template authors can do?
- 5 "The surround" is everything on a page outside the operative content object. Why is separating the surround from the content object such a useful idea? **ANALYZE**
- 6 The course claims "truly understanding a CMS often means understanding how it aggregates." Why would aggregation, of all things, be the deepest test of a system? **DEBATE**
- 7 Aggregations vary along many axes (static vs. dynamic, manual vs. derived ordering, etc.). For a homepage "featured" section, which settings would you choose? **APPLY**
- 8 "Organizing content often creates content." Do you agree that a curated list deserves the same management (versioning, permissions) as the items it groups?
- 9 The "tyranny of the tree" describes too much functionality bound to one hierarchy. Have you seen a system where one structure controlled too much? **REFLECT**
- 10 When the same content must appear in two places in a tree, systems use a "main" location plus a reference. What problems can this create for editors? **ANALYZE**
- 11 Content is consumed inside "cognitive containers" like pages or app screens. How does designing for the container change how you model the pieces? **APPLY**
- 12 As delivery spreads across web, email, app, and voice, does the classic "page" remain a useful unit — or is it becoming obsolete? **DEBATE**

CMS Architecture — Coupled, Decoupled & Headless

Focus: The management and delivery environments, the three coupling models, how publishing differs, the historical swing, and the case for decoupling.

- 1 Every CMS has a management side and a delivery side. Why is the relationship between these two the "most significant" architectural decision? **ANALYZE**
- 2 In a coupled system, publishing is "almost anticlimactic" — just flipping a flag. In a decoupled one, it means moving data. How does that change how teams think about "going live"?
- 3 Compare coupled, decoupled, and headless in your own words. Which would you default to for a brand-new small business site, and why? **APPLY**
- 4 The default architecture swung decoupled → coupled → decoupled again. What does this reveal about how technology trends actually move? **REFLECT**
- 5 "Vendors move in packs" as external technology shifts. Is this herd behavior good or bad for customers? **DEBATE**
- 6 Decoupling can improve security by shrinking the delivery server's attack surface. How much should security concerns alone drive an architecture choice? **DEBATE**
- 7 Decoupling lets you avoid licensing a CMS on every delivery server. When do cost savings like this outweigh the added operational complexity? **APPLY**
- 8 Headless "waits to be pulled" rather than "pushing" content out. What kinds of projects most benefit from that inversion of control? **ANALYZE**
- 9 A bank's custom platform can't be "owned" by a CMS, forcing the CMS to be subservient. Where else might content be secondary to a system's primary purpose? **APPLY**
- 10 The deciding question is "how many channels must you serve?" Is channel count really the best single predictor of the right architecture? **DEBATE**
- 11 The course warns against buying "architectural complexity you don't need." How can a team tell the difference between future-proofing and over-engineering?
- 12 If headless maximizes flexibility but requires building your own front end, who should *not* choose headless? **APPLY**

Acquiring & Living With a CMS

Focus: The ecosystem, the selection funnel, the out-of-the-box myth, governance, content migration, "software is not your savior," and the future of the field.

- 1 A CMS ecosystem (developers, integrators, docs, community) is called a "core feature." How would you actually evaluate the health of an ecosystem before buying? **APPLY**
- 2 "There is no soulmate for your project, and all that glitters will eventually lose its shine." How do you set realistic expectations without becoming cynical? **REFLECT**
- 3 The selection process is a "funnel of deepening analysis." What criteria belong at the wide top of the funnel versus the narrow bottom? **ANALYZE**
- 4 Out-of-the-box functionality is often "illusory." Why do buyers keep falling for the OOTB promise despite repeated disappointment? **DEBATE**
- 5 "Poor governance and vague ownership do far more damage than a lack of technology." Do you agree that people problems outweigh tool problems? **DEBATE**
- 6 "Software is not your savior." How would you push back on a leader who believes a new platform will fix a struggling website? **APPLY**
- 7 Developers are famously drawn to *building* a CMS, but the course says resist. When, if ever, is building your own genuinely justified?
- 8 Migration is called maybe "the most dangerous word in CMS." Which of its phases (extract, transform, reassemble, import, resolve, QA) seems riskiest to you, and why? **ANALYZE**
- 9 "Content velocity" means content keeps changing even while you migrate. How would you manage a migration of a site that never stops publishing? **APPLY**
- 10 CMS software is described as very "sticky." Is high switching cost a sign of a good investment or a trap to avoid? **DEBATE**
- 11 "Launch day is not the finish line, it's the starting line." What ongoing commitments does an organization take on the day *after* launch? **REFLECT**
- 12 Among the predicted trends (consolidation, decoupling's return, marketing focus, SaaS at the low end, multichannel, distributed intake), which do you find most convincing today, and which has aged poorly? **DEBATE**