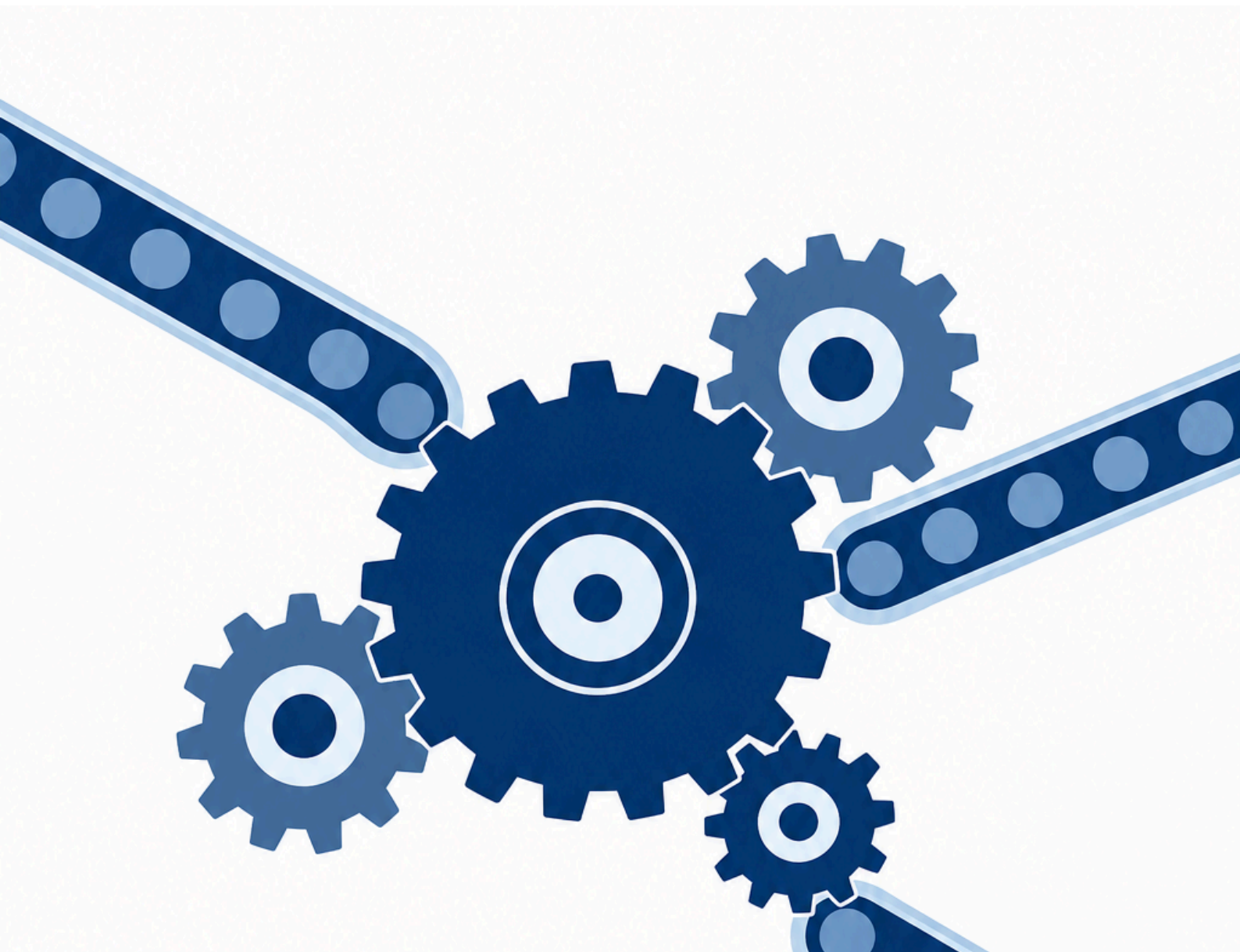


The Content Engine

How content teams build systems that publish every week without burning out

THE UNPLAIN TEAM



Contents

The content engine model	3
The five parts and their owners	4
Why production starves the rest	4
Burnout as a systems signal	5
Roles and the workflow	5
Three roles that carry the engine	5
The pipeline	6
The bottleneck and the weekly review	8
Repurposing and distribution	9
Derivatives, not fragments	9
Owned, earned and paid	10
The channels that matter for a small team	10
Measurement	11
The limits of attribution	11
Automation and operating cadences	12
The cadences	13
A twelve-week rollout	13
References	14

Executive summary

Most content teams do not fail on quality. They fail on consistency. A team that can produce an excellent article once a quarter and cannot produce a good one every week has a systems problem, not a talent problem, and hiring does not fix a systems problem. This report describes the system: a content engine with five parts, named roles, a staged pipeline, a short list of metrics and a few rituals that keep it running when individuals are on vacation, sick or simply tired.

The argument in brief: the unit of work is the brief, not the draft; publish less than you can produce and spend the difference on distribution; keep measurement short and honest; automate handoffs, never judgments. The advice assumes a team of two to eight people.

The content engine model

The word engine is deliberate. An engine converts inputs into motion in a repeatable way, regardless of who is standing next to it. A content program that depends on one person's energy is not an engine; it is a performance, and performances end.



Five parts, one machine: inputs, production, distribution, measurement and feedback.

The engine has five parts. Inputs are the ideas, research, expert access and audience understanding that give the team something worth saying. Production turns an accepted idea into a finished piece. Distribution puts it in front of the people it was written for. Measurement reads what happened. Feedback changes the next cycle in response. Donella Meadows describes a system as elements, interconnections and a purpose, and notes that its behavior is a product of its structure rather than of its parts [1]. The parts of a content engine are easy to list; the interconnections are where teams fail.

The five parts and their owners

Part of the engine and its owner	The question that part answers	How it fails when nobody owns it
Inputs, owned by the head of content	What is worth saying, and to whom?	Pitches come from whoever shouts loudest
Production, owned by the managing editor	Can we ship it on time at the bar we set?	Drafts pile up in editing and quality drifts
Distribution, owned by the distribution lead	Will the right people actually see it?	Publish and pray, with one post per channel
Measurement, owned by the analyst or the editor	What happened, and what does it mean?	Vanity dashboards that nobody reads
Feedback, owned by the head of content	What do we change in the next cycle?	Metrics are reviewed and nothing changes

The owners are roles, not people; a six-person team will have people holding two roles each. A part with no owner does not get done. Inputs and feedback share an owner on purpose: the person who decides what to cover should be the person who sees what happened last time.

Why production starves the rest

Drafts and published pieces are tangible; distribution and measurement produce nothing that looks like output, so when the calendar slips they are cut first, and the engine runs at full capacity while delivering less each month, because nobody is reading the work and nobody knows it. Donald Reinertsen's work on product development flow explains the mechanism: queues grow sharply as utilization approaches full capacity, and a system with no slack cannot absorb variability [2]. The practical rule is to plan production at roughly seventy to eighty percent of measured capacity and give the rest to distribution, updating and surprises.

A team that publishes every week has not found more time; it has removed the decisions that used to consume it.

Burnout as a systems signal

Christina Maslach and Michael Leiter describe burnout as exhaustion, cynicism and a sense of inefficacy [3]. Each maps onto a failure in the engine: a calendar with no buffer, publishing into silence, and no feedback on what worked. The fix for the first is a buffer of two to four finished, unpublished pieces, which turns a weekly deadline from a cliff into a conveyor.

Roles and the workflow

Editorial operations is the discipline of deciding who decides. Roles are not job titles. They are sets of decisions, and a decision with no owner gets made late, by committee or by whoever is most stressed at the time.

Three roles that carry the engine

The managing editor holds the calendar, runs the pipeline review and is the only person who can add a piece to the calendar, move a deadline or kill a piece. Executive requests go through the same gate: they may be prioritized, but they are not inserted. A team that cannot kill work accumulates half-finished drafts that drain attention for months.

The managing editor's job is not to make the team write more; it is to make sure that everything the team writes was worth starting.

The editor is the role most often collapsed into the writer, and that is the collapse to resist: nobody publishes a piece that only they have read. Subject matter experts are borrowed from product, engineering or sales, and they belong at the brief stage. An expert who agreed to the angle rarely rejects the draft; one who first sees a finished piece will rewrite it.

The pipeline

The pipeline is production made explicit: a sequence of stages, each with an entry criterion, an exit criterion and an owner. Without criteria a piece can sit in editing for a week while the editor and the writer each believe the other is holding it.



Each stage has an entry rule and an exit rule, so nothing sits between owners.

Stage and its typical turnaround	What must be true before work enters	What must be true before work leaves
Pitch and prioritize, at the weekly meeting	An idea is logged with its intended reader and a one-line angle	The pitch is accepted against a quarterly theme, parked or killed, and an accepted pitch has a slot and a writer
Brief, one to two days	A slot and a writer are assigned	The editor has signed off the brief and the expert has agreed to the angle
Draft, three to five working days	The brief is signed off	The writer has checked the draft against the brief and the checklist
Edit, two days	The draft is submitted with its sources	The editor marks it ready or returns it once with consolidated notes
Expert review, two to three days	The edited draft has its claims flagged	Every flagged claim is confirmed or corrected
Design, two to three days	The text is locked	Assets are approved and alt text is written
Publish, same day	Assets and metadata are complete	The piece is live, checked and logged
Distribute, over two to four weeks	The piece is live	Every planned derivative is scheduled with an owner

Two rules carry most of the weight. The brief names the reader, the outcome, the one claim the piece makes, the evidence, what the piece is not about, the expert and the derivatives planned, and it determines the quality of everything downstream. The edit allows one return; a draft that would need a second return has a brief problem and goes back to the brief.

The bottleneck and the weekly review

Every pipeline has one stage that limits its throughput, and Eliyahu Goldratt's observation holds for content as well as for factories: improving any stage other than the bottleneck improves nothing [4]. In content teams the bottleneck is almost always editing, because the editor is also writing, managing or both. The remedy is to protect editing time and cap work in progress with a limit on each column of the board. When the edit column is full, writers repurpose or update older pieces instead of starting new drafts.

The weekly pipeline review takes twenty minutes and follows a fixed script.

- 1** Walk the board from right to left, starting nearest publication, and ask of each piece what it needs to exit and who is doing that today.

- 2** Give every piece that has outstayed its typical turnaround a decision: unblock, reassign or kill.

- 3** Check each column against its limit; if a column is over, nothing new enters until it is under.

- 4** Confirm the buffer; if fewer than two finished pieces are waiting, refilling it is next week's priority.

- 5** Log the decisions in one place as the input to the monthly performance review.

Repurposing and distribution

Distribution is the part of the engine that most teams underfund. Producing a piece feels like progress; distributing it feels like promotion, and many content people got into content because they did not want to do promotion. A piece nobody reads has the same return as a piece nobody wrote, and a cornerstone piece deserves at least as much effort in distribution as it took to produce.

Derivatives, not fragments

Repurposing is producing several pieces from one source asset. Done well it is the highest-leverage activity in the engine, because the expensive part, the research and the argument, has already been paid for. Done badly it is cutting an article into pieces and posting the pieces, which the audience recognizes as recycling. A derivative is a new piece, written natively for a different channel, that carries one idea, stands on its own and links back to the source with context. The richest sources are research reports, recorded talks and customer interviews, and the derivative schedule is planned with the source, not after it.



A derivative is a new piece written for its channel, carrying one idea from the source.

A derivative is a new piece with an old source, not an old piece in a new format.

Owned, earned and paid

Owned channels are the ones the team controls: the website, the newsletter, its social accounts. Earned channels are carried by other people: search, staff and customers sharing, press. Paid channels buy reach and stop the day the budget stops. Les Binet and Peter Field distinguish long-term brand effects, which build slowly and persist, from short-term activation effects, which spike and decay, and argue that both are needed in balance [5]. Owned and earned channels do the long-term work, and paid activates specific pieces for specific audiences at specific moments.

The channels that matter for a small team

Effort is stated as effort rather than money, and time to signal is the interval before a team can tell whether a channel is working at all.

Channel and how the team holds it	What the channel is good for	Effort required and time to signal
Email newsletter, an owned channel	Returning readers, direct replies and a home for every derivative	Medium effort every week; a signal within days
Website or blog, an owned channel	The permanent home for every piece and the feed for search	Medium effort, mostly upfront; a signal within months
Organic search, an earned channel	Durable traffic on evergreen topics that compounds for years	High effort upfront, low ongoing; three to six months to a signal
Staff profiles on LinkedIn, an earned channel	Trust, and far more reach than a company page can get	Low effort per post but needs genuine buy-in; a signal within days
Guest articles and partner co-marketing, earned channels	Borrowed and adjacent audiences, links and relationships	Medium effort per piece, relationship-led; a signal within weeks
Paid social and newsletter sponsorships, paid channels	Amplifying pieces already proven on owned channels	Low effort but a budget is needed; a signal within days

The table is a menu, not a to-do list. A team of six can run a newsletter and one social channel well, let search grow from the website, and hold paid in reserve for proven pieces. Podcasts, video and webinars are not worthless, only expensive for a small team. Staff profiles are the most underused channel: a company page is read as advertising, a person's post is read as a person.

Measurement

Measurement fails in one of two directions: page views alone, which answer no useful question, or a dashboard of forty metrics that changes no decision. The remedy for both is a short list of metrics, each with a definition, a calculation anyone could do by hand and the question it answers. Avinash Kaushik's distinction helps: metrics are numbers, and key performance indicators are the subset tied to a business objective [6]. The table runs from operational metrics, fully in the team's control, through audience metrics to business metrics.

Metric and how to calculate it	What the metric describes	The question it answers
Publishing consistency: on-time pieces divided by planned slots	Share of planned slots filled on time	Is the engine reliable?
Cycle time: median days from brief sign-off to publication	How long a piece really takes	Where is the waiting?
Rework rate: twice-returned drafts divided by all drafts	Share of drafts returned more than once	Are briefs and edits working?
Reach: sessions or views summed across channels	People who saw a piece	Did distribution work?
Engaged read rate: completions divided by views	Share of readers who reached the end	Is the piece worth reading?
Subscriber growth: signups minus unsubscribes each month	Net new newsletter readers	Are we earning return readers?
Assisted conversions: paths that include a content page	Conversions where content was one touch	Does content appear in the journey?
Internal use: shares and links counted in the CRM and support tickets	Pieces used by sales or support	Is content useful beyond marketing?

The first three rows are the ones most teams never measure, and they predict the rest: a team with a high rework rate will not fix its audience metrics by changing topics. If only one business metric can be kept, keep internal use, because a salesperson who sends an article to a prospect has judged that it helps.

The limits of attribution

Every attribution method is a model, not a measurement. Last-click gives all the credit to the final touch and undercounts content, which tends to appear early in a journey; Google's own documentation is explicit that its models credit the same path differently [7]. Sharing in chats arrives

as direct traffic, and the effects of good content are delayed by months [5]. Report figures with their model named, compare periods on the same model, and treat them as one input to a judgment.

If a metric cannot change what the team does next week, it is a report, not a measurement.

The monthly performance review reads the sheet in tier order and records exactly three decisions: one thing to do more of, one thing to stop and one experiment. A review that produces ten decisions produces none.

Automation and operating cadences

The right tool stack is smaller than most teams own: a board, shared documents, a content management system, a scheduler, an analytics view and templates. Every tool should own one stage or one handoff. Automation makes a handoff happen without anyone remembering it: the board moves a card when a status changes, the scheduler posts the derivative on the planned day.

AI drafting tools are good at transformation and poor at judgment. Turning a transcript into prose, an article into candidate posts or a draft into alt text are transformations. Deciding what a piece argues, whether a claim is true and whether it should be published are judgments, and a tool's draft enters the pipeline at the draft stage like any other.

Automate the handoffs and keep the judgments; a tool that is handed a judgment returns content nobody is accountable for.

The most reliable automation is the template: a brief that forces the writer to name the reader and the claim, and checklists run before submitting and after publishing. Atul Gawande's case for checklists is that they catch the steps experts skip under pressure [8].

The cadences

A cadence is a ritual held at a fixed interval whether or not anyone feels the need for it.

- Daily and asynchronous: a short message from anyone who is blocked, and nothing from anyone who is not.
- Weekly, as two separate meetings: the editorial meeting decides pitches and priorities; the pipeline review inspects stage ages and limits. Merging them debates pitches while the stuck drafts stay stuck.
- Every two weeks: the distribution sync plans derivatives and channels for upcoming pieces.
- Monthly: the performance review logs its three decisions, and a half-day refresh pass updates or retires the pieces that carried the most search traffic last quarter.
- Quarterly: planning sets the two or three themes the team will own, and the retrospective fixes the system rather than the people, with every change given an owner.

A twelve-week rollout

The plan assumes a team that already publishes something and can name a managing editor. The order matters: the pipeline and the brief come before the buffer, the buffer before the cadence and the cadence before measurement, because measuring an engine that is not yet running produces noise.

Weeks and what the team installs	Who owns each step	What done looks like
Weeks one and two: audit pieces, tools and roles, then define the pipeline and build the board	The head of content, then the managing editor	A one-page inventory is shared and the board is live with its stage criteria written on it
Weeks three and four: adopt the brief template and agree the roles	The editor, then the head of content	Three real briefs are signed off and every named owner has agreed in person
Weeks five and six: build the buffer and start the weekly cadence	The writers, then the managing editor	Two pieces are finished and unpublished, and two editorial meetings have been held on time
Weeks seven and eight: launch the distribution checklist and instrument the metrics	The distribution lead, then the analyst	Every live piece has a distribution plan and the metric sheet is filled by hand

Weeks and what the team installs	Who owns each step	What done looks like
Weeks nine and ten: run the first repurposing cycle and automate two handoffs	The distribution lead, then the managing editor	One rich source has become four derivatives, and publishing and tagging run without a reminder
Weeks eleven and twelve: hold the first performance review, then the retrospective and quarter plan	The head of content, then the whole team	Three decisions are logged and next quarter has themes, slots and owners

Week five is the hard week. The team keeps publishing at its existing pace while also finishing two pieces it does not publish. That is the only way to build a buffer without a publishing gap, and teams that cannot absorb it should extend the plan by a week rather than skip the buffer.

The team does not get faster; it stops doing the work that was never going to be read.

By week twelve the team publishes from a buffer and has made the three decisions it was previously too busy to make.

References

- [1] Meadows, D., Thinking in Systems: A Primer, Chelsea Green Publishing, 2008.
- [2] Reinertsen, D., The Principles of Product Development Flow: Second Generation Lean Product Development, Celeritas Publishing, 2009.
- [3] Maslach, C. and Leiter, M., The Truth About Burnout: How Organizations Cause Personal Stress and What to Do About It, Jossey-Bass, 1997.
- [4] Goldratt, E. and Cox, J., The Goal: A Process of Ongoing Improvement, North River Press, 1984.
- [5] Binet, L. and Field, P., The Long and the Short of It: Balancing Short and Long-Term Marketing Strategies, Institute of Practitioners in Advertising, 2013.
- [6] Kaushik, A., Web Analytics 2.0: The Art of Online Accountability and Science of Customer Centricity, Sybex, 2009.
- [7] Google, About attribution and attribution modeling, Google Analytics Help documentation, accessed 2026.
- [8] Gawande, A., The Checklist Manifesto: How to Get Things Right, Metropolitan Books, 2009.

Publish the report, not the formatting

Write it in Markdown. Unplain lays out the tables, pull quotes and contents.

Create a PDF free