



Kentico

#2343	Change color of the..	Resolved
#2432	Small text change	Pending
#2428	Button color	Pending
#2567	Small text change	Pending



EBOOK

Quick One? A Field Guide to Working Without the Ticket Queue.

kentico.com

What's Inside.

Quick one?	3
The Ticket Hall of Fame.	4
Chapter 1: Find where the time really goes	5
Chapter 2: Four things marketers should own outright	8
More numbers worth knowing.	14
Chapter 3: When you should still call a developer	15
Chapter 4: Where AI fits	17
Chapter 5: Control, not autonomy theatre	19
Chapter 6: The 30-day autonomy plan	21
About Kentico.	23

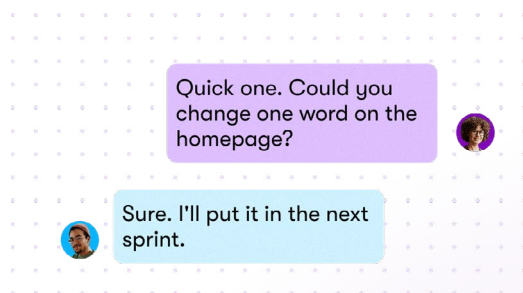
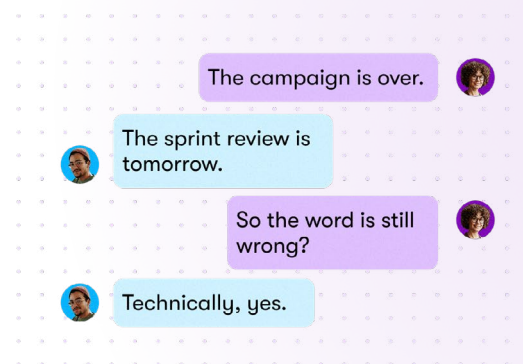
Quick one?

Every marketer knows the sentence: “Quick one, could you change a word on the homepage?” Every developer knows the answer: “Sure, it will be in the next sprint.”

Two weeks later the word is still wrong, the campaign has moved on, and two perfectly good professionals are annoyed with each other. Nobody did anything wrong, the system just made a small correction too slow.

This guide is about ending that, and it is not a story about marketers versus developers. Marketers get to stop writing 20 tickets a month. Developers get to stop receiving 20 “quick ones” a month and go back to the work they were hired for. Everyone moves faster, and the Friday afternoon mood improves for both teams.

You will find real customer stories, a few numbers that surprised us, a 2-minute self-test, and a 30-day plan you can start on Monday.



The Ticket Hall of Fame.

Before the numbers, here's a few scenes. If you work with a website, you've experienced at least three of them.

- **The “quick one:”** Someone from marketing walks over, smiles, and opens with those two words. The developer has already started sighing. Nothing in the next ten minutes will be quick.
- **The one-word change:** A typo on a landing page. One word. Estimated delivery: two weeks, because it needs a ticket, a slot, a review and a release.
- **The button color saga:** Week three of waiting for a button to turn from blue to orange. The campaign it was meant for has ended. The button is now orange for a campaign that does not exist.
- **“It worked yesterday:”** The sentence that makes marketing and development look at each other across the room with exactly the same expression.
- **The ownership question:** Sprint planning, someone asks who owns “just updating the website”, and the room discovers a new kind of silence.
- **The plot twist:** A marketer builds a landing page alone, with no supervision, and publishes it. The developers watch, braced for something to break. Nothing breaks.

The last scene is the point of this guide. Autonomy is not a power grab; it's a way to give two teams their time back, helping them move faster, execute more effectively, and deliver better business outcomes.

Your turn. Which scene is yours? Send us your own “ticket hall of fame” moment and the best ones may end up in the next edition.



Chapter 1:

Find where the time really goes

Most teams cannot say how much of their week disappears into waiting, because waiting doesn't show up on any calendar. Two minutes with the audit below will show you.

The dependency audit. Write down the last 10 things your team wanted to publish or change on the website. For each one, mark whether it needed a developer, and why.

#	What we wanted to do	Needed a developer?		Real reason
1	Fix a typo or swap an image	☐ yes	☐ no	
2	Publish a new landing page	☐ yes	☐ no	
3	Change a headline to test it	☐ yes	☐ no	
4	Add a form to a campaign	☐ yes	☐ no	
5	Update a price or a date on several pages	☐ yes	☐ no	
6	Reorder the sections of a page	☐ yes	☐ no	
7	Launch an email campaign	☐ yes	☐ no	
8	Add a new region or language page	☐ yes	☐ no	
9	Connect a form to the CRM	☐ yes	☐ no	
10	Change a button color or label	☐ yes	☐ no	

Adjust the rows to match your own last ten, then count your “yes” answers.

0 to 2 yes: you already have real autonomy. Chapter 5 will help you protect it.

3 to 6 yes: you are paying a hidden tax. Most of those “yes” answers are habit, not necessity. Chapters 2 and 3 sort the two apart.

7 to 10 yes: you are stuck in the ticket queue.
Start with the 30-day plan in Chapter 6.

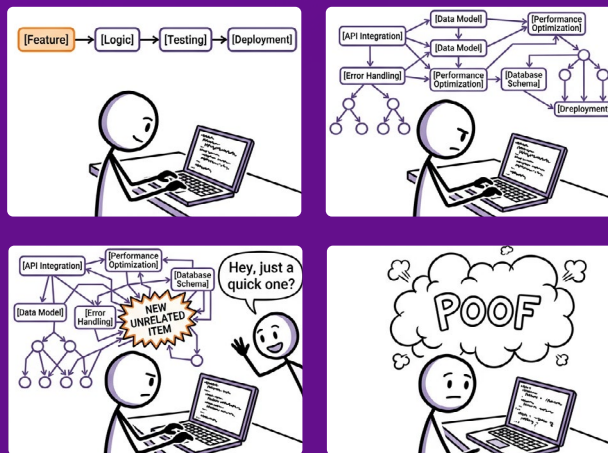
Now look at the “real reason” column. For each “yes”, ask: did this need a developer’s judgment, or just a developer’s permissions?



Fun fact: the 25-minute detour.

Research from UC Irvine found it takes [around 25 minutes to get back to an interrupted task](#). A [2024 Duke and Vanderbilt study](#) found interruptions measurably disrupt reading, writing and reviewing code. So a “quick one” is rarely quick for the person receiving it.

Do the math (illustrative). 20 tickets a month at about 24 minutes of refocus each is roughly 8 hours of lost concentration per developer per month, before anyone has written a line of code for the ticket itself. That is a full working day, spent switching back and forth.

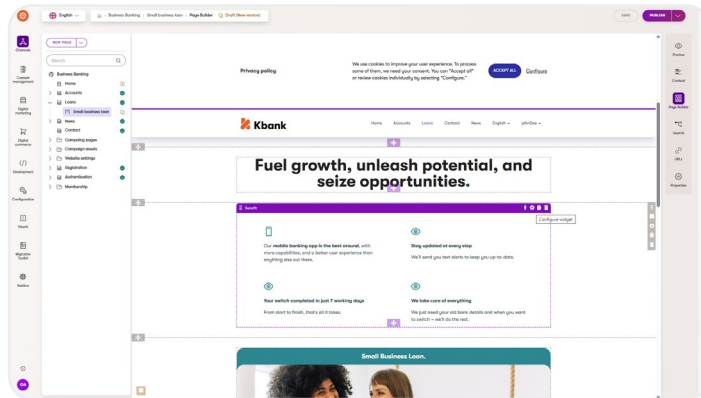


Sources: [How much is waiting on developers costing your marketing team?](#). The 8-hour figure is our own arithmetic on the cited research, not a measured result.

Chapter 2:

Four things marketers should own outright

Owning something means you can start it, change it and ship it without opening a ticket. Developers still build the foundation once: the components, the content types, the permissions. Marketers then use them as often as they like. Here are the four areas where that split pays off fastest.



1. Pages

What owning it looks like: you assemble landing pages from approved, on-brand components, preview them, and publish. A new page takes an afternoon, not a sprint.

The story. UniCredit Bank, working with ACTUM Digital, moved away from a CMS where every specialized page needed extensive development. The marketing team built its own campaign page in Xperience by Kentico and A/B tested it against the old product page. Click-through on “Open Account” went from 12% to 25.4% in under two months.

12% to 25.4%. Click-through on “Open Account” after the marketing team built and tested its own page. In under two months.

What developers get: no more “can you build us a page for this?” Their time goes to the components that make every future page possible.





Warning
Campaign events are not automatically published and explicated based on the campaign start and end date. You need to manually ensure that page and content from events are published as needed.

Name	Asset type	Channel	Content type	Actions
AAA Northampton County	Contact form	-	Product - Loan	[E] [X]
AAA Northampton County	Page	Personal Banking	Product page	[E] [X]
AAA Northampton County	Contact form	-	Product - Loan	[E] [X]
AAA Northampton County	Page	Personal Banking	Product page	[E] [X]
AAA Northampton County	Form	-	-	[E] [X]
AAA Northampton County	Automation process	-	-	[E] [X]
AAA Northampton County	Automation process	-	-	[E] [X]
AAA Northampton County	Automation process	-	-	[E] [X]
AAA Northampton County	Page	Personal Banking	Marketing Page	[E] [X]
AAA Northampton County	Form	-	-	[E] [X]
AAA Northampton County	Page	Marketing Service	Product page	[E] [X]
AAA Northampton County	Contact form	-	Product - Loan	[E] [X]



3. Campaigns

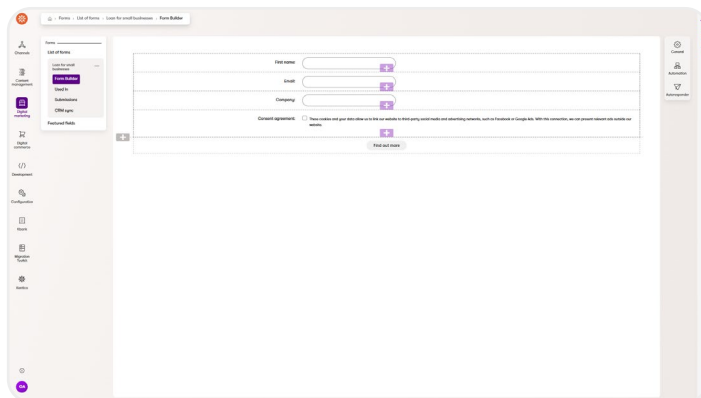
What owning it looks like: the email, the page it links to and the audience live on the same platform, so you launch in hours and not after three handoffs.

The story. AAA Northampton County, with Citro Digital, upgraded from an aging platform that made email campaigns slow to build. With native email marketing, contact groups and reusable components, the team sent 550,000 emails in its first three months. Centenary University streamlined its campaigns on Xperience by Kentico and saw engagement rise 40%.

550,000 emails in the first three months for AAA Northampton.
40% more engagement at Centenary University.

What developers get: no more template fixes and list exports for each send.





4. Forms

What owning it looks like: you build the form, choose where it lives and change a field when the campaign changes. Developers connect the form to your systems once, and you handle everything in front of that connection.

The story. Cockburn ARC uses Page Builder and Form Builder so its team creates pages and forms on its own, faster than before.

What developers get: the form requests that used to arrive on a Thursday with “needed by Friday” attached.

“

I don't have to send an email to my web developer and wait for my request to get into their production schedule. I have all the tools at my fingertips to edit my website so that it becomes more than just the digital nametag for our company.”



Kevin Collins.

Marketing Manager, Palmer Holland



More numbers worth knowing.

-75%

less developer time for
Visit New Hampshire.

-50%

less development time
for Racing Queensland.

+90%

improvement in workflow time at
United Federal Credit Union.

+84%

more engagement at Beacon Bank
after it built a library of flexible
components with ZAG Interactive.

250+

franchise microsites consolidated
into one solution by Bluebird
Care, with Ridgeway. The project
won Kentico Site of the Year 2024
in healthcare.

*These are customer-reported results from their own projects.
Your numbers will differ, but the direction tends to be the same.*



Chapter 3:

When you should still call a developer

Marketing autonomy doesn't mean developers are optional. It means they stop spending their days on work that does not need their judgment, so they have time for work that does. Calling a developer isn't a failure of autonomy. It is the system working as designed.

A useful test: does this need a developer's judgment, or only a developer's permissions? If it is only permissions, the platform should give them to you. If it's judgment, pick up the phone.

Keep these with a developer:



New building blocks. A new component, widget or content type. Developers build it once, properly, and you reuse it for years.



Integrations. Connecting the website to your CRM, ERP or any system where data moves between tools.



Security, performance and data protection. Anything involving authentication, personal data or how fast the site loads.



Data models. How content is structured behind the scenes, because everything you build later depends on it.



Upgrades, migrations and architecture decisions. The foundation work that keeps everyone else fast.

Hand these to marketers:



page building, editing and publishing content, campaign setup, and forms , using the components and workflows developers have approved.

Devs, this one is for you. Every ticket you do not receive for a typo is a morning you keep for the architecture work you truly enjoy. Your name is on the foundation. Marketers just get to build on it, and they'll be grateful that you built it well.

A developer's honest list. This is the place for a partner or MVP voice: a short, candid list of what they still insist on handling themselves, and why. We are collecting these from the Kentico community.

The credibility check for this chapter is simple. A guide that claims everything can be self-served is selling something. One that says where the line sits is trying to help you.



Chapter 4: Where AI fits

AI is the third “teammate” in the room, and the question is whether it makes the ticket queue shorter or just moves it somewhere new. Our answer: AI helps most when it takes the first draft and leaves the decision to you.

Agents collaborate with marketers. They propose; marketers review, approve, and launch.



That sentence is the whole model. An agent can suggest a page structure, draft copy, or point out where a customer journey stalls. A marketer reads it, changes it, approves it and publishes it. Nothing goes live without a human saying yes.

What this changes in your week:

- The blank page disappears. You start from a draft, not from nothing.
- Analysis arrives with a suggestion attached, so you spend your time deciding instead of digging.
- Developers are not asked to translate every idea into a ticket first.

What does not change:

- accountability. You still own what goes live, and the people who built your components still own the foundation.

This is also why marketing autonomy is not the same thing as autonomous AI. The first is about who controls digital experiences day to day. The second would be about taking people out of the loop. We aren't interested in the second.

Rule of thumb.

The most useful question to ask about any AI feature is simple: who approves it before it reaches a customer? If the answer is “nobody”, it's not autonomy. It's a risk.



Chapter 5: Control, not autonomy theatre

Autonomy theatre is when a team is told it can publish anything, then discovering every page still needs three approvals and a developer's blessing. Everyone acts out of independence while the queue carries on backstage. Real autonomy comes with real control, built into how workflows.

Three things make the difference:



Permissions.

Decide who can edit, who can publish and who can only suggest. Give people the access that matches their responsibility, and no more.



Workflows.

Define the path a page takes from draft to live, so it's clear who reviews what and in what order. A good workflow is a short, visible path, not a maze.



Approvals.

Keep a human sign-off where it matters, such as legal review on regulated content or brand review on homepages, and skip it where it doesn't.

The test for each rule you add: does it protect the brand or the customer, or does it only protect someone from having to trust someone else? Keep the first kind. Question the second.

When governance is designed this way, developers stop acting as the gatekeepers of every change, because the guardrails live in the platform. Marketers move fast inside lines everybody agreed on, and nobody holds their breath at publish time.

One question to take to your next planning meeting.
Who owns “just updating the website”? If the room goes quiet,
you have found your first workflow to design.



Chapter 6: The 30-day autonomy plan

You don't need a transformation program. You need four weeks, one honest list and a developer who is happy to help. Tick the boxes as you go.

Week 1:

See it

- Run the dependency audit from Chapter 1 on your last 10 requests.
- Count your tickets for the last month and note how many were "only permissions" requests.
- Share the results with your developers and ask them which three tickets they would happily never see again.

Week 2:

Agree on the line

- Use the Chapter 3 test to sort your requests into "marketers own" and "developer judgment".
- Agree on who owns pages, content, campaigns and forms, in writing.
- Pick the one component or template developers should build first, because you will reuse it the most.

Week 3:

Try it

- Build and publish one page yourself, end to end, using approved components.
- Set up one workflow and one approval step that you actually need, and remove one you do not.
- Ask an agent for a first draft of something, review it, edit it and decide.

Week 4:

Measure it

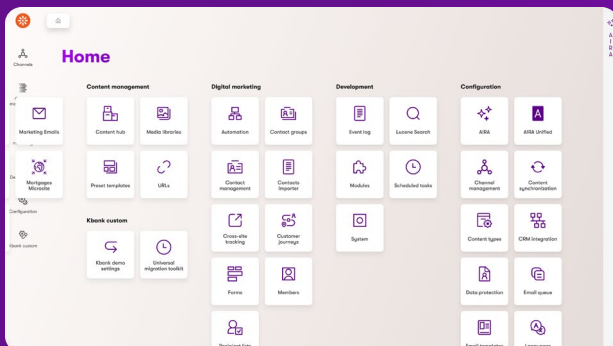
- Count your tickets again and compare with Week 1.
- Ask developers what they did with the time.
- Choose the next three requests to move out of the queue, and repeat.

The only success metric that matters in month one: the number of tickets nobody had to file. Take a screenshot of the first week it drops. Your developers will want to frame it.



About Kentico.

Kentico is a content management system with built-in digital marketing, native commerce, and AI agentic capabilities that help deliver personalized customer experiences through websites, microsites, emails, apps, digital kiosks, and other channels. It replaces multiple disconnected tools with a unified solution that reduces technology complexity and improves team productivity.



Empower your team to create engaging experiences while eliminating time-consuming tasks with a comprehensive platform that is easy to learn and use, powered by built-in AI agentic capabilities that streamline marketing workflows. Tailor content, commerce experiences, and delivery to individual preferences and develop consistent interactions across multiple digital touchpoints—reaching your customers on their favorite channels, any time, anywhere. Kentico consolidates multichannel content management, digital marketing, and commerce capabilities in a single solution, helping your team achieve more with fewer processes and less technology.

→ MEET OUR EXPERTS

H.Q

Kentico software s.r.o.
Nové sady 996/25
602 00 Brno
Czech Republic

CZ

Kentico software s.r.o.
FLEKSI BETA
Beta Building
Vysokálova 1481/4
140 00 Praha 4-Michle

US

Kentico Software, LLC
15 Constitution Drive,
Suite 2C
Bedford, NH 03110
United States

UK

Kentico Software Ltd
One London Square
Cross Lanes
Guildford, Surrey,
GU1 1UN
United Kingdom

APAC

Kentico Software Pty Ltd.
83 Mount St, Level 4
North Sydney, NSW 2060
Australia

Germany

Kentico Software GmbH
c/o Schnorbus Helmholtz
Wardemann Park GmbH
Kanalstraße 2
41460 Neuss



kentico.com