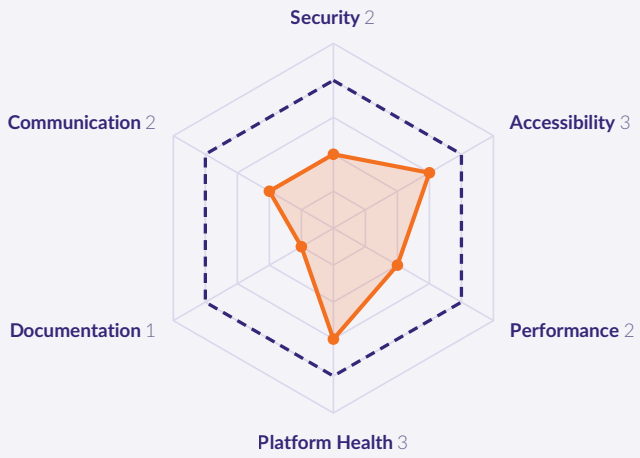
Bright Plum

THE WEB OPERATIONS RADAR

Is your website an asset or a liability?

When web operations go well, nothing happens. No outage, no breach, no complaint letter. That's hard to put in a budget request. This rubric gives you something you can put in one: six scores, each tied to a written description anyone can check.

Security training teaches everyone the same habit. Before you trust a message, check the domain it came from. For a prospective student, a parent or a donor, that domain leads to your website. Six areas of web operations decide whether the site keeps that trust or breaks it.



How to read the radar

Each spoke is one dimension, scored 1 to 5. The further a point sits from the center, the healthier that dimension.

The solid shape is where a site stands today. The dashed shape is the target: level 4 on every spoke.

A lopsided shape is normal. It usually means a two- or three-person team can only cover so much. The widest gap between the two shapes is where the risk concentrates, and it's usually the place to start.

Today Target (level 4) Example site

The six dimensions

Security. Keeping the site patched, backed up and access-controlled, and being able to recover it when something goes wrong.

Performance. How fast pages load for real visitors on their phones, and whether anyone measures and defends that speed as the site changes.

Documentation & Institutional Knowledge. Whether what it takes to run the site is written down and shared, or held in one person's head.

Accessibility. Whether people using assistive technology can use the site, and whether anyone tests that. It's also a legal obligation for colleges and universities.

Platform Health. Whether the software under the site (CMS core, PHP, plugins or modules, hosting, deployment) is current, supported and on a plan.

Communication & Reporting. How web operations reach leadership: what gets reported, what gets escalated, and whether the website has a seat at planning.

What each level looks like

Find the row, then read left to right. A dimension earns the highest level where that cell and every cell to its left are true.

Dimension	1 · Ad hoc	2 · Reactive	3 · Defined	4 · Proactive	5 · Managed
Security	No update schedule. CMS core and plugins run behind known-vulnerable versions. No one owns patching.	Updates happen when something breaks or a vendor scare hits the news. No staging environment, so changes go straight to production.	• A monthly patch cycle runs. Backups run, but nobody has tested a restore. SSL and access control get an occasional look.	Patching on a tested schedule through staging. Restore drills prove the backups work. Access audited each semester.	Monitoring and alerts reach a person. A written incident-response plan has been rehearsed. Security status goes to leadership.
Accessibility	Never audited. Missing alt text, keyboard traps and contrast failures across templates. Legal exposure unknown.	Fixes happen only after a complaint or an OCR letter. Knowledge lives with one person. New content brings old errors back.	• A full-site scan against WCAG 2.2 AA in the last 12 months. A known-issues list exists, but fixes go unfunded and slip.	Automated and manual testing on every release. Authors trained. Templates produce accessible pages by default.	Accessibility built into procurement, design and publishing. The conformance statement is current and holds up.
Performance	No one measures. Pages ship with uncompressed images and unused scripts. Mobile experience unknown.	Speed gets attention when someone important complains. One-off fixes wear off within a semester.	Core Web Vitals tracked for key pages. Image and caching standards exist, but the CMS doesn't enforce them.	Speed limits set for each template. Slowdowns caught before launch. CDN and caching tuned and written down.	Real-visitor speed measured across the site, tied to enrollment-funnel metrics and reviewed each quarter.
Platform Health	CMS at end of life or unsupported. Plugin and module sprawl. Abandoned subsites nobody can name an owner for.	Upgrades put off until forced. Dev and staging environments missing or out of step with production.	• CMS core and PHP within vendor support. Technical debt tracked informally. Content sprawl known but ungoverned.	Upgrade roadmap funded and scheduled. Every subsite inventoried with a named owner. Debt paid down each cycle.	Platform decisions follow a written architecture plan. Stale sites get retired on a schedule.
Documentation & Institutional Knowledge	Everything lives in one person's head. Passwords sit in inboxes. One departure would strand the website.	Scattered notes in old tickets and personal drives. A new hire spends months digging for how things work.	• Access verified and credentials in a shared vault. Core runbooks exist but go stale. Some knowledge still lives only in heads.	Documentation reviewed on a schedule. A capable new hire or vendor could take over within weeks.	The knowledge base is the source of truth. Every recurring task has a runbook and every system has an owner.
Communication & Reporting	Leadership hears about the website only when it's down. No metrics and no shared words for risk.	Updates go out when someone asks. Numbers pulled by hand and framed differently each time.	• A report goes out on schedule. It lists tickets closed, and leadership can't tell what changed.	A regular scorecard ties web health to institutional goals. Risks reach leadership early, with options attached.	Web operations has a seat at planning. Leadership makes budget calls from current data it trusts.

• Shaded level 3 cells contain a requirement that holds the whole dimension at 2 until it's met. Page 3 lists them. Performance has none.

SCORING

How a site earns its score

The grid shows what each level looks like. These rules decide which level a site gets.

1. No skipping levels

A dimension scores at the highest level where everything at that level and below is true. A team doing excellent level 5 monitoring with one level 3 gap scores 3. That isn't a penalty. It points at the exact gap to close.

We saw this with a client this year. A year of real accessibility fixes, and the score held, because the follow-up scan that confirms those fixes hadn't been approved yet. The work was done. The evidence wasn't there, and the score said so.

2. Six requirements hold a dimension at 2

Miss one and the dimension can't reach level 3, whatever else is done. Each one covers a risk nothing else does.

Dimension	Must be true	Why it caps
Security	Security updates applied in the last 45 days	Published vulnerabilities come with working attack code
Security	Backups configured, with a known retention window	Nothing else survives a total loss
Accessibility	A full-site scan in the last 12 months	You can't fix what nobody measured
Platform Health	CMS core and PHP within vendor support	Out of support means no patches exist to apply
Documentation	Access verified, credentials in a shared vault	If nobody can get in at 2 a.m., the rest is theory
Communication	The report goes out on schedule	If leadership never sees the work, it doesn't exist to the people funding it

Performance has none, on purpose. A slow site is bad, but it isn't dangerous. If every dimension had a cap, most charts would stay red and nobody would read them.

3. Show progress inside the level

Every score carries a count, like "Accessibility: 3. One of four requirements met toward level 4." The score can sit still for a month while the count moves. That keeps a good month from looking like a flat one.

4. Recurring work expires

A monthly task counts for 45 days, a quarterly one for 120, an annual one for about 13 months. The grace period covers a late report. It doesn't cover a skipped quarter.

5. "Not applicable" needs a reason

Some requirements don't fit some sites. A site with no third-party integrations doesn't need a review of them. Write down why, and the item drops out of the count. No reason, no exclusion: a blank reason looks the same as a hidden gap. The six requirements above can't be marked not applicable.

6. Aim for 4

Level 5 depends on things only the institution controls: its incident-response plan, its procurement process, whether leadership shows up to planning. A good team can get most sites to 4. Nobody gets to 5 alone.

7. Score each site on its own

An institution with three websites has three sets of scores. When you need one number, use the weakest site and name it.

START HERE

Four questions to ask without a developer

You don't need a scan to start. Ask your team these and listen for one thing: is the answer a document, or a name?

DOCUMENTATION

If the person who runs our site went on vacation tomorrow and something broke, would we know what to do? Where is that written down?

ACCESSIBILITY

If general counsel got an accessibility complaint about the website, what would we show them? Who has the record of what's been fixed?

SECURITY

If a serious vulnerability were announced tomorrow, how would we know whether it affects us, and who would patch it?

PLATFORM HEALTH

If the site went down, how would we restore it, and how long would it take? Has anyone tested that?

If an answer is a person's name instead of a process, that dimension is probably a 1 or a 2. That's a fine place to start. It's also the first line of your next budget request.

Bring it to your budget conversation

Most higher-ed web budgets get built in January and February and decided by March or April. Score your site before then. Six numbers, each tied to a description your VP can check, make a stronger case than a list of tickets closed.

We use this rubric in our own client reports, and we revise it as we learn. Use it, share it, adapt it.
Bright Plum, Inc. · Web operations for higher-ed teams on Drupal and WordPress · brightplum.com