

Healthcare software has a strange kind of gravity. People do not just “use” an EHR. They depend on it during time pressure, with incomplete information, while working around pain, fatigue, and distractions that are not going away. Accessibility in that context is not a box-checking exercise. It is a design commitment that affects safety, speed, accuracy, and dignity.

When EHR accessibility is done well, it disappears into the workflow. Clinicians move through tasks without wrestling with focus traps, confusing tab order, unlabeled controls, or dense pages that effectively punish anyone with a cognitive or visual impairment. When it is done poorly, the experience can become hostile even for staff without disabilities, because usability problems stack. One bad interaction creates workarounds. Workarounds create mistakes. Mistakes create risk.

This article focuses on practical design decisions that improve both usability and inclusivity in electronic health record systems, with real constraints: legacy screens, vendor customizations, busy clinical teams, and the fact that accessibility touches everything from HTML semantics to how medication directions are phrased.

Accessibility is not separate from usability

A common misconception is that accessibility is a narrow set of technical requirements. In reality, most accessibility improvements are usability improvements under a different lens.

Consider a “small” issue like a search field without a proper label. Sighted users may still infer intent from placeholder text, but screen reader users will not. In an EHR, the result can be more than inconvenience. If staff cannot quickly locate a patient’s prior lab results, they may repeat orders or miss a critical trend. Even for staff without assistive technology, a poorly labeled interface often signals a larger pattern: inconsistent controls, ambiguous navigation, and unclear task structure.

Accessibility and usability also overlap in how people process information. EHR screens often pack multiple data types into a single view: demographics, vitals, problems, medications, allergies, immunizations, orders, messages, and billing-relevant fields. When the visual hierarchy is weak, cognition takes the hit. If a clinician has to re-read the same row labels or hunt for the “active” item, the interface is not just “hard,” it is exhausting. Exhaustion is a human factor, and the people affected by it include everyone, not only those with diagnosed disabilities.

The best accessible EHR design supports multiple paths through the same task: keyboard navigation works, screen reader output makes sense, visual contrast is adequate, and layout reduces the amount of mental glue work required to interpret dense information.

Where EHRs get tricky: workflow, not just screens

It is easy to evaluate an EHR one screen at a time. In practice, the experience is a chain of micro-interactions. Accessibility gaps often appear at transitions: opening a chart, drilling into an order, switching tabs, editing a form, confirming a change, and returning to the list.

Here are a few high-risk moments I have seen cause real trouble in clinical settings:

- Moving between interactive regions without a predictable focus order.
- Expanding “details” sections that are not announced to assistive technology.
- Using color alone to convey status (like warning, completed, or overdue).
- Requiring fine motor control for tasks that could be structured more robustly.

- Hiding key information behind collapsed UI that is not reachable efficiently by keyboard.

The core accessibility question is not “does this screen meet a guideline.” It is “can the user complete the job under pressure using tools they rely on, without losing context.” A usable interface helps everyone, and an accessible interface prevents the workflow from breaking when someone’s interaction method differs.

Keyboard access and focus management in real workflows

In many EHR environments, keyboard navigation is not optional. Some clinicians prefer it because it is fast. Others use it because mouse control is impaired by fatigue, [electronic health record interoperability](#) tremor, or workstation setup. For screen reader users, keyboard control is the bridge to all other interaction.

Good keyboard accessibility in an EHR goes beyond letting users press Tab. It includes:

- Focus order that matches the visual and logical order.
- Visible focus indicators that do not vanish against the background.
- Clear focus placement when dialogs open, and returning focus when dialogs close.
- Preventing focus traps in modals or expandable panels.
- Ensuring that expandable content behaves consistently (so it is not “there,” but unreachable).

One practical lesson: focus behavior can differ across browsers and across custom components. A team might “fix” focus in one area and accidentally regress it elsewhere when a new widget is added. In an EHR, regression happens fast because releases are frequent and UI components are reused.

A pattern I like for teams is to define focus rules per component type, then test those rules across the most important navigation paths: patient chart entry, order entry, documentation editing, and message review.

Screen reader output: labels, structure, and meaningful feedback

Screen reader accessibility often fails for the same reason many usability issues persist: developers focus on visual appearance and assume the meaning is obvious. For screen reader users, meaning must be explicit.

The “obvious” part that gets missed is structure. A screen reader needs more than individual controls. It needs a meaningful page outline, headings that reflect task structure, and form fields with clear accessible names.

In an EHR, accessible names matter because clinical terms are specific. A button labeled “Submit” is rarely useful when a user needs to understand whether they are submitting an order, signing a note, or confirming medication instructions. Labels should communicate the action plus the context.

Feedback is equally important. If a user saves a note and nothing appears to happen, the screen reader user needs a clear announcement: “Saved successfully” or “Order updated” or “Validation error, field X.” Without that feedback, users may repeat actions or assume the system is stuck.

Testing for screen reader output should include scenarios with validation errors. Error states are where many interfaces become opaque. For example, a required field might be marked with a red outline visually, but the screen reader may not receive a descriptive error message. In a clinical workflow, that can delay documentation or medication entry, which is exactly when users are most time constrained.

Color, contrast, and status: beyond making text “visible”

EHR interfaces commonly use color to convey status: pending, discontinued, critical, completed, or flagged. Color can be helpful, but it becomes exclusionary when it is the only cue.

Good inclusive design handles status through multiple channels:

- Text that includes a status word, not just a color.
- Icons paired with accessible labels.
- Consistent meaning for each visual cue across the entire product.
- Sufficient contrast for both foreground and background elements, including borders and focus indicators.

There is a trade-off here. Designers often try to reduce visual noise by using lighter styling. In an EHR, less contrast can be attractive aesthetically, but it can also slow down scanning. I have watched clinicians take longer to find critical items when the interface uses subtle gray distinctions. Accessibility improvements, especially around contrast and clear status labeling, often pay off in speed for everyone.

Form design: reducing cognitive load without removing necessary detail

EHR forms carry the weight of clinical documentation: allergies, problem lists, medication orders, history, assessment, and plan. Some forms are long. Some require conditional fields. Some allow free text where clinicians must capture nuance, but also must avoid ambiguity.

Accessibility intersects with cognitive load in multiple ways:

- Too many fields on one screen create overwhelm.
- Ambiguous field labels force users to guess.
- Inconsistent ordering between similar forms increases mental effort.
- Unclear units and formatting increase the risk of mistakes.
- Autocomplete and suggestion lists can become confusing if they do not behave predictably.

A design that is accessible often looks like a design that respects attention. For instance, grouping related fields and using headings helps screen reader users navigate and helps sighted users scan. But grouping needs to be consistent. If grouping differs from one form to another, users lose their “mental template.”

Conditional logic needs careful handling too. If selecting one option reveals new fields, the interface must communicate that change to assistive technology and ideally keep focus in a predictable location. Otherwise, keyboard users can miss newly available inputs or not realize that validation requirements have changed.

Language, terminology, and the reality of diverse users

Accessibility is not only about disability categories. EHR users come from different training backgrounds, and many are multilingual. Even when the interface is in one language, the content may include abbreviations and clinical jargon.

If an EHR uses shorthand that assumes insider knowledge, it becomes harder for new staff and for users who do not share the same familiarity with local conventions. That issue is not always captured by typical accessibility testing, but it affects usability and inclusivity.

Inclusive language choices include:

- Avoiding overly cryptic field labels when a clearer phrase exists.

- Keeping instructions consistent across the application.
- Using plain, unambiguous terms for validation messages.
- Providing examples for complex inputs like dosing frequency, especially when multiple formats are accepted.

A practical example from a documentation flow: if the interface asks for “sig” without explaining what kind of content is expected, new users either enter inconsistent text or rely on guesswork. That inconsistency can propagate into downstream tasks like medication reconciliation.

When teams improve terminology and validation messages, clinicians often adopt the system faster. They also make fewer formatting errors, which means fewer corrections and less rework.

Making the EHR navigable: information architecture matters

EHR navigation is often structured around tasks, tabs, and grids. Grids are notoriously hard for accessibility if not designed thoughtfully, especially when users need to sort, filter, select rows, and edit inline.

Good navigability means users can:

- Understand where they are and what area of the chart they are interacting with.
- Find the next step without hunting.
- Return to the same location they left when they open details.
- Use consistent navigation patterns across modules.

In practice, a lot of accessibility issues come from inconsistent patterns. One module may place the primary action button at the top, while another module places it at the bottom. One module may use expandable panels, another uses separate pages, and a third uses dialogs. Each choice adds variation that keyboard users and screen reader users must learn repeatedly.

Teams can reduce this burden by standardizing component behaviors: consistent headings, consistent dialog focus rules, consistent action placement, and consistent announcements for dynamic updates.

Testing accessibility without drowning in tooling

Accessibility testing should include more than automated checks. Automated tools can catch missing labels and some structural issues, but they cannot fully validate real workflow usability. In an EHR, the “correctness” of output depends on how the system behaves during task completion, including validation timing and focus transitions.

A strong testing approach usually combines:

- Automated audits for baseline issues.
- Screen reader walkthroughs for core flows.
- Keyboard-only sessions for the same flows.
- Short usability sessions with diverse users, including users who rely on assistive technology and users who are new to the product.

Also, do not limit testing to a single patient record layout. Real charts contain variations: missing data, long medication lists, unusual lab values, and documents with different states. Those differences often reveal accessibility and usability edge cases that clean demo patients never show.

If your team has limited time, prioritize testing around high-impact tasks. In an EHR, tasks that involve ordering, documenting, and communicating results should be treated as accessibility-critical.

A quick design review checklist for accessibility in EHR UI

- Ensure every interactive control has a clear accessible name that matches the clinician’s task context
- Verify keyboard-only navigation works end-to-end, including dialogs and expandable sections
- Confirm focus is managed predictably when content changes dynamically
- Make status visible through text or icons, not color alone
- Test error messages and validation states with screen reader and keyboard interaction

This checklist is intentionally short. In practice, the value comes from applying it consistently to the most important workflows and iterating after real usage.

Inclusive design for clinicians with limited motor or visual ability

Accessibility work often focuses on screen readers and contrast, but EHR interactions can also exclude users with motor impairments, limited range of motion, or difficulty with precise cursor control.

Some interface patterns to examine:

- Small click targets and tightly spaced controls.
- Hover-only interactions that require precise cursor placement.
- Drag-and-drop that cannot be replaced with keyboard alternatives.
- Auto-advancing UI that steals focus while the user is still interacting.
- Fine-grained sliders for dosing or selection instead of structured inputs.

Replacing fragile controls with robust alternatives does not mean removing power. It means offering input methods that match different needs. For example, where the interface uses a dropdown for medication selection, it should support keyboard filtering and provide clear selection confirmation. Where it uses date pickers, the keyboard navigation should be predictable and the input format should be tolerant.

For vision impairments, larger text support and zoom behavior are critical. Users may zoom the browser or increase system-level text size. If layouts break at higher zoom, fields can overlap, content can become inaccessible, or horizontal scrolling becomes required in ways that interrupt workflow.

In the EHR context, zoom behavior can be the difference between “workable” and “unworkable.” A clinician under time pressure cannot be asked to fight layout issues.

Handling dynamic content: when the chart updates while you work

EHRs are interactive systems. Content changes when orders are placed, when documents are signed, when results come in, and when alerts fire. Accessibility becomes complex when dynamic updates occur without clear communication.

A robust accessible design should ensure that updates are either:

- Notified in a way assistive technologies can detect and announce, or
- Delayed until the user reaches a stable point in the workflow, so they are not interrupted mid-entry.

For instance, if a saving action triggers a refresh that resets focus to the top of the page, keyboard users can lose their place. Screen reader users may re-encounter the same content repeatedly, which can cause confusion or delays.

Teams can reduce harm by preserving user state across updates. If a user is editing a section, updates should not yank focus. If the system detects a conflict or validation issue, it should show it clearly where the user can address it.

Dynamic content also includes things like loading indicators. A spinner without an accessible announcement can be a dead end for screen reader users. Even for sighted users, a spinner with no progress description can create doubt, leading to repeated clicks or navigation away from the page.

Two common failure patterns teams can spot early

1. Decorative visuals are treated as content. The UI shows meaning through layout or color, but the underlying semantic structure does not convey that meaning to assistive technology.
2. The “happy path” works, but task interruption breaks accessibility. Dialogs open without focus changes, validation errors are not announced, and dynamic updates reset the user’s place.

These patterns often show up early in a build, which is good news. The earlier a team identifies them, the cheaper it is to fix the foundations. Later fixes frequently require reworking shared components and regression testing across multiple modules.

The trade-offs you will face, and how teams make judgment calls

Accessibility is not free of trade-offs. In EHR product development, decisions happen under constraints like performance, legacy compatibility, and UI complexity.

Here are a few trade-off areas that come up frequently:

- **Performance versus richer announcements.** More announcements and more semantic updates can increase complexity and potentially impact responsiveness. The right approach is to announce only what matters for user decision-making.
- **Consistency versus local workflow needs.** Some departments develop conventions that differ from the default. Total consistency across the entire EHR may not be realistic, but accessibility behaviors like focus management and labeling should stay consistent.
- **Minimal UI versus discoverability.** Hiding controls behind icons or progressive disclosure can reduce clutter, but it can also hide key functions from keyboard and screen reader users if the interaction model is not solid. The goal is progressive disclosure with strong accessibility semantics.
- **Customization versus QA coverage.** EHRs often support configuration. If accessibility is implemented in one configuration path but not in others, users will still hit gaps. Teams need a strategy to test enough configurations to reduce risk.

My practical advice: treat accessibility like safety. You cannot test everything. You can, however, define what “good” looks like for core interactions and ensure the design system enforces those behaviors. When the product grows, the enforcement mechanism keeps accessibility from drifting.

Practical next steps for product teams and implementation partners

Even the best EHR design can be undermined by implementation choices, configuration, and training. Accessibility should be treated as a shared responsibility across product teams and the organizations deploying the system.

A solid starting point is to prioritize a few core workflows and commit to measuring whether users can complete them efficiently with different interaction methods. Keyboard-only success, screen reader comprehension, and time to complete high-risk tasks are concrete outcomes that matter.

Equally important is documentation and accountability. If engineers know what “focus should return to the triggering element” means, the system is more likely to behave correctly across components. If designers know what makes a validation error message helpful, the entire interface becomes clearer.

Finally, involve clinicians and support staff in the conversation. A feature that seems accessible in a lab can still fail in the real environment, where users are switching between multiple systems and communicating with patients and colleagues. Real feedback often points to the most damaging friction points, like confusing status meanings or error messages that do not translate into next actions.

Accessibility in an EHR is not a one-time deliverable. It is a discipline, and it improves as teams get better at observing how people work when the stakes are high.

Why inclusive accessibility pays off beyond compliance

Accessibility can be justified by ethics, by legal requirements, and by design principles. But in healthcare software, there is also a practical reason it matters: it reduces avoidable friction during moments when people need clarity.

When an EHR respects multiple ways of interacting, it helps:

- clinicians who rely on keyboard navigation to move fast,
- staff who use screen readers to interpret dense information,
- users with limited fine motor control who need reliable target sizes and predictable focus,
- patients and caregivers who interact with portals built on similar accessibility principles.

The common theme is dignity. Accessible design signals that the system expects users to be different and still be able to complete their tasks.

If you build an EHR that works under pressure, for more people, in more circumstances, you are not just improving usability. You are designing for safer care and a more humane workflow.