

**Analiza grešaka DynUI projekta i plan rešavanja**

**Izvršni rezime**

Projekat DynUI ima **78+ TypeScript grešaka** raspoređenih u **13 komponenti**, sa **prosečnim odstupanjem od funkcionalne specifikacije od 62.7%**. Analiza je pokazala da glavni uzroci problema nisu tehnički već organizacioni - nedosledno praćenje funkcionalne specifikacije, odsustvo Code Review procesa i neimplementirani Quality Gates iz CI/CD pipeline-a.[[1]](#fn1)[[2]](#fn2)[[3]](#fn3)[[4]](#fn4)

**Kategorizacija grešaka**

Greške su podeljene u četiri glavne kategorije:

**1. Nedostajući API propovi (45% grešaka)**

* Props interfejsi nisu potpuni prema specifikaciji
* Nedostaju kontrolni propovi (value, defaultValue, onChange)
* Nedostaju pristupačnost propovi (aria-label, aria-labelledby, data-testid)

**2. Nekonzistentni tipovi (30% grešaka)**

* Tipovi nisu centralizovani u src/types/ modulu
* Module exports nisu pravilno postavljeni (TS2305, TS2724 errors)
* Type compatibility problemi između komponenti i hook-ova

**3. Neimplementirani design patterns (20% grešaka)**

* Controlled/Uncontrolled pattern samo delimično implementiran
* Compound Components bez Context API
* WAI-ARIA patterns nisu implementirani
* ForwardRef nedostaje u nekim komponentama

**4. Test infrastruktura (5% grešaka)**

* Miksovani jest/vitest framework-ovi
* Accessibility testovi ne rade (missing axe-core setup)
* Type definicije za testove problematične[[1]](#fn1)

**Kritične komponente - detaljni pregled**

**DynStepper (11 grešaka, 85% odstupanje)**

**Problem**: API potpuno odstupa od specifikacije. Nedostaju svi kontrolni propovi definisani u DynUI-FS-02.[[2]](#fn2)

**Specifične greške**:

* Props interfejs ne sadrži: value, defaultValue, onChange, id, aria-label, aria-labelledby, data-testid
* Tipovi nisu eksportovani: DynStepperRef, DynStep, DynStepperProps
* label property ne postoji na StepItem tipu
* onChange callback nije pravilno tipiziran (async vs sync problem)

**Rešenje**:

1. Definisati kompletan DynStepperProps interfejs prema specifikaciji
2. Kreirati i eksportovati DynStepperRef i DynStep interfejse
3. Implementirati Controlled/Uncontrolled pattern koristeći useState
4. Dodati label property na StepItem tip
5. Osigurati da onChange vraća sinhroni callback (ne Promise)

**Trajanje**: 3-4 dana | **Prioritet**: VISOK

**DynMenu (10 grešaka, 80% odstupanje)**

**Problem**: Compound Component pattern nije implementiran. Context API nedostaje što uzrokuje prop drilling.[[2]](#fn2)

**Specifične greške**:

* Props interfejs ne sadrži: items, orientation, id, aria-label, aria-labelledby, data-testid, onAction
* DynMenuItem nije pravilno eksportovan (pogrešan naziv u importu)
* MenuItem tip zahteva label čak i za divider type
* Test greška: focus property ne postoji na Element type

**Rešenje**:

1. Proširiti DynMenuProps sa svim propovima iz specifikacije
2. Ispraviti export - eksportovati MenuItem (ne DynMenuItem)
3. Kreirati union type za MenuItem koji dozvoljava divider bez label property
4. Implementirati WAI-ARIA Menu pattern sa role="menu", menuitem, aria-expanded
5. Dodati tastaturnu navigaciju (Arrow keys, Enter/Space, Escape)
6. Kreirati MenuContext za deljenje state-a između parent i child komponenti

**Trajanje**: 4-5 dana | **Prioritet**: VISOK

**DynListView (10 grešaka, 90% odstupanje)**

**Problem**: Komponenta potpuno odstupa od Form input kategorije. Svi kontrolni propovi nedostaju.[[4]](#fn4)[[2]](#fn2)

**Specifične greške**:

* Props interfejs ne sadrži: items, value, defaultValue, multiSelect, disabled, onChange, aria-label, aria-labelledby
* Test greške: jest namespace errors, implicit any types na callback parametrima
* Selectable List pattern nije implementiran

**Rešenje**:

1. Redefinisati DynListViewProps prema Form input kategoriji iz specifikacije
2. Dodati obavezne propove: items (array), value, defaultValue, multiSelect, disabled, onChange
3. Implementirati selection state management (single/multi select)
4. Dodati aria-\* atribute za pristupačnost (role="listbox", aria-multiselectable)
5. Implementirati tastaturnu navigaciju (Arrow keys, Home/End, Space/Enter)
6. Dodati tipizaciju za callback parametre (ukloniti implicit any)
7. Refaktorisati testove sa jest na vitest

**Trajanje**: 5-6 dana | **Prioritet**: VISOK

**DynTabs (7 grešaka, 75% odstupanje)**

**Problem**: WAI-ARIA Tabs pattern nije potpuno implementiran. DynTabItem tip nema value property što uzrokuje višestruke greške.[[2]](#fn2)

**Specifične greške**:

* Props interfejs ne sadrži: value, orientation, activation, fitted
* DynTabItem.value property nedostaje (pojavljuje se u 15+ error lokacija)
* onChange nije pravilno tipiziran (inconsistent signature)

**Rešenje**:

1. Proširiti DynTabsProps sa: value, orientation, activation, fitted
2. Dodati value property na DynTabItem tip
3. Ispraviti onChange signature: (value: string) => void
4. Implementirati potpuni WAI-ARIA Tabs pattern sa role="tablist", role="tab", role="tabpanel"
5. Dodati aria-selected, aria-controls, aria-labelledby atribute
6. Implementirati tastaturnu navigaciju (Arrow keys, Home/End)
7. Ispraviti axe accessibility testove

**Trajanje**: 3-4 dana | **Prioritet**: VISOK

**Ostale komponente**

**DynToolbar (6 grešaka, 60% odstupanje)**

* Nedostaju propovi: children, id, aria-\*, data-testid
* Ne prihvata ref prop - potreban forwardRef
* **Trajanje**: 2-3 dana | **Prioritet**: SREDNJI

**DynTable (6 grešaka, 55% odstupanje)**

* Nedostaje sortable prop
* TableSortState objekat nema key property
* DynTableColumn nema header property
* onSort callback prima samo jedan parametar umesto dva
* **Trajanje**: 2-3 dana | **Prioritet**: SREDNJI-VISOK

**DynBox (7 grešaka, 70% odstupanje)**

* Props interfejs ne sadrži: padding, radius, background, align, justify, direction, wrap
* Style Props pattern nije konzistentno primenjen
* **Trajanje**: 1-2 dana | **Prioritet**: SREDNJI

**DynDivider (5 grešaka, 100% odstupanje)**

* Svi tipovi nedostaju u centralnom types modulu
* Module exports nisu postavljeni
* **Trajanje**: 1 dan | **Prioritet**: NIZAK-SREDNJI

**index - Root exports (5 grešaka, 100% odstupanje)**

* Nedostaju eksporti: ThemeProvider, useTheme, IconDictionaryProvider, classNames, generateInitials
* Provider pattern nije eksportovan
* **Trajanje**: 2-3 dana | **Prioritet**: VISOK

**Analiza design pattern-a**

**Pattern Compliance - trenutno stanje**

|  |  |  |  |
| --- | --- | --- | --- |
| Design Pattern | Status | Compliance Rate | Kritični problemi |
| **Controlled/Uncontrolled** | ⚠️ DELIMIČNO | ~40% (4/10) | DynListView, DynStepper, DynTabs nedostaju value/defaultValue propovi[[2]](#fn2)[[4]](#fn4) |
| **Compound Components** | ❌ NE | 0% (0/4) | DynMenu, DynTabs - Context API nije implementiran[[2]](#fn2) |
| **Forward Ref** | ⚠️ DELIMIČNO | ~70% (7/10) | DynToolbar ne prihvata ref; DynInput ref nekonzistentan[[2]](#fn2)[[4]](#fn4) |
| **WAI-ARIA Practices** | ❌ NE | ~20% (1/5) | Aria atributi nedostaju; tastaturna navigacija nije implementirana[[2]](#fn2)[[3]](#fn3)[[4]](#fn4) |
| **Validation Pattern** | ⚠️ PROBLEM | ~50% | DynInputValidationRule nije kompatibilan sa hook interfejsom[[4]](#fn4) |
| **Style Props Pattern** | ⚠️ DELIMIČNO | ~50% | DynBox koristi propove koji nisu u interfejsu[[2]](#fn2) |

**Preporuke za unifikaciju implementacije**

**1. Controlled/Uncontrolled Pattern - Unified Implementation**

Trenutno stanje: Nedostaju value i defaultValue propovi u većini form komponenti.[[4]](#fn4)[[2]](#fn2)

// Preporučeni pattern  
const [internalValue, setInternalValue] = useState(defaultValue)  
const currentValue = value !== undefined ? value : internalValue  
  
const handleChange = (newValue) => {  
 if (value === undefined) {  
 setInternalValue(newValue)  
 }  
 onChange?.(newValue)  
}

**2. Compound Components - Context Pattern**

Trenutno stanje: Props se prosleđuju manuelno što uzrokuje prop drilling.[[2]](#fn2)

// Preporučeni pattern za DynMenu  
const MenuContext = createContext<MenuContextValue>(null)  
  
export const DynMenu = ({ children, onAction, orientation }) => {  
 const [activeItem, setActiveItem] = useState(null)  
   
 return (  
 <MenuContext.Provider value={{ activeItem, onAction, orientation }}>  
 <div role="menu">{children}</div>  
 </MenuContext.Provider>  
 )  
}  
  
export const DynMenuItem = ({ value, label }) => {  
 const { activeItem, onAction } = useContext(MenuContext)  
 // ...  
}

**3. WAI-ARIA Implementation**

Trenutno stanje: Aria atributi nedostaju u Props interfejsima.[[3]](#fn3)[[4]](#fn4)[[2]](#fn2)

Za svaku komponentu dodati:

* Props: aria-label, aria-labelledby, aria-describedby, data-testid
* Role atribute prema WAI-ARIA Authoring Practices
* Tastaturnu navigaciju (Arrow keys, Home/End, Enter/Space/Escape)
* Focus management sa useRef i focus() metode

**4. Type Centralization**

Trenutno stanje: Tipovi nisu centralizovani, što uzrokuje TS2305 greške.[[4]](#fn4)

src/  
 types/  
 common.types.ts # Shared types (Size, Variant, Color)  
 component.types.ts # Per-component types  
 index.ts # Central export

**Plan rešavanja grešaka**

**Faza 1: Kritične greške i struktura (Sedmica 1-2, 8 dana)**

**Zadatak 1: Centralizacija tipova i exports (3 dana)**

* Komponente: index, DynDivider
* Deliverables:
  + src/types/ sa svim tip definicijama
  + src/theme/ sa ThemeProvider i useTheme
  + src/icons/ sa IconDictionaryProvider
  + src/utils/ sa classNames i generateInitials
  + Ažuriran src/components/index.ts
* DoD: Svi TypeScript module errors (TS2305, TS2724) rešeni. Build prolazi bez grešaka[[3]](#fn3)[[4]](#fn4)

**Zadatak 2: Implementacija Form Input komponenti (5 dana)**

* Komponente: DynListView, DynInput, DynTreeView
* Deliverables:
  + Potpuni Props interfejsi sa svim obaveznim poljima
  + Controlled/Uncontrolled state management
  + Validation pattern implementiran
  + onChange callbacks sa pravilnim signaturama
* DoD: Svi "Property does not exist" errors rešeni. Unit testovi prolaze. Validation radi[[2]](#fn2)[[4]](#fn4)

**Faza 2: Kontrole i navigacija (Sedmica 3-4, 9 dana)**

**Zadatak 3: Refaktorisanje navigacionih komponenti (7 dana)**

* Komponente: DynStepper, DynMenu, DynTabs
* Deliverables:
  + Potpuni Props sa value, onChange, aria-\* atributima
  + WAI-ARIA role i state management
  + Tastaturna navigacija (Arrow keys, Home/End, Enter/Space/Escape)
  + Focus management
* DoD: Svi type errors rešeni. Axe accessibility testovi prolaze. Tastaturna navigacija funkcionalna[[3]](#fn3)[[4]](#fn4)[[2]](#fn2)

**Zadatak 4: Toolbar i Page komponente (2 dana)**

* Komponente: DynToolbar, DynPage
* Deliverables:
  + forwardRef implementacija
  + Prošireni Props interfejsi
  + DynButtonKind sa danger variantom
* DoD: ref prop radi. DynButtonKind type errors rešeni[[4]](#fn4)[[2]](#fn2)

**Faza 3: Layout i prikaz podataka (Sedmica 5, 4 dana)**

**Zadatak 5: Layout komponente (2 dana)**

* Komponente: DynBox
* Deliverables:
  + DynBoxProps sa padding, radius, background, align, justify, direction, wrap
  + Style Props pattern dokumentovan
  + Storybook primeri
* DoD: Svi layout props funkcionalni. Props dokumentovani u Storybook[[2]](#fn2)[[4]](#fn4)

**Zadatak 6: Data Display komponente (2 dana)**

* Komponente: DynTable
* Deliverables:
  + DynTableColumn sa header property
  + TableSortState sa key property
  + onSort callback sa dva parametra
  + aria-sort atributi
* DoD: Sortiranje funkcionalno. Aria attributes pravilno postavljeni[[4]](#fn4)[[2]](#fn2)

**Faza 4: Testovi i kvalitet (Sedmica 6, 5 dana)**

**Zadatak 7: Migracija test infrastrukture (3 dana)**

* Komponente: Sve komponente sa test greškama
* Deliverables:
  + vitest.config.ts konfigurisano
  + @axe-core/vitest ili vitest-axe instaliran
  + toHaveNoViolations matcher registrovan
  + Svi testovi refaktorisani
* DoD: Svi testovi prolaze. Axe testovi bez serious/critical violations[[3]](#fn3)[[4]](#fn4)

**Zadatak 8: Finalna provera i dokumentacija (2 dana)**

* Komponente: Sve
* Deliverables:
  + tsc --noEmit bez grešaka
  + eslint bez errors
  + Svi unit testovi ≥80% coverage
  + Storybook dokumentacija kompletna
  + API kontrakti dokumentovani
* DoD: Svi Quality Gates prolaze. Release notes spremne[[3]](#fn3)[[4]](#fn4)

**UKUPNO TRAJANJE**: 26 radnih dana (~5-6 sedmica)

**Resursi i rizici**

**Potrebni resursi**

* **Frontend Developer (Senior)**: 1 FTE, 6 sedmica
* **A11y Specialist (Consultant)**: 0.2 FTE, 2 sedmice
* **Tech Lead (Review)**: 0.3 FTE, 6 sedmica
* **QA Engineer**: 0.5 FTE, 2 sedmice (Faza 4)

**Rizici**

|  |  |  |
| --- | --- | --- |
| Rizik | Nivo | Mitigacija |
| Breaking changes u API-ju mogu uticati na postojeće konzumere | ⚠️ VISOK | Kreirati deprecation notices; verzionisati API prema SemVer; komunikovati promene unapred[[2]](#fn2) |
| Nedostaje vizuelna regresiona test infrastruktura | ⚠️ SREDNJI | Postaviti Chromatic/Playwright u Fazi 4; manual review u međuvremenu[[3]](#fn3) |
| Scope može rasti ako se otkriju dodatni problemi | ⚠️ SREDNJI | Time-box svaki zadatak; prioritizovati kritične greške; defer nice-to-have features[[3]](#fn3) |

**Root Cause Analysis**

Analiza je identifikovala **pet glavnih uzroka** problema:

**A. Nepoštovanje funkcionalne specifikacije**

* Specifikacija DynUI-FS-01/02/03 jasno definiše API, ali nije praćena
* Props interfejsi implementirani bez reference na spec dokumente
* Nedostaje traceability između spec → impl → test[[2]](#fn2)[[4]](#fn4)

**B. Nedostatak Code Review procesa**

* API changes nisu reviewed prema specifikaciji
* TypeScript errors nisu blokirali merge
* Nedostaju PR checkliste iz DynUI-FS-03[[3]](#fn3)[[4]](#fn4)

**C. Neadekvatna CI/CD pipeline**

* Quality Gates (A-F) iz spec-a NISU implementirani
* TypeScript compilation nije required za merge
* A11y testovi nisu automatizovani (axe-core missing)[[3]](#fn3)[[4]](#fn4)

**D. Nekonzistentna implementacija**

* Svaka komponenta implementirana drugačije
* Nedostaju shared utilities i base patterns
* Duplikacija koda umesto reusable patterns[[4]](#fn4)[[2]](#fn2)

**E. Nedostatak dokumentacije i onboarding-a**

* Developeri ne znaju za postojanje func spec dokumenata
* Nedostaje contribution guide sa pattern examples
* Nema centralnog mesta za design decisions[[2]](#fn2)[[3]](#fn3)

**Uticaj na Definition of Done**

Prema DynUI-FS-03, komponenta je završena kada ispunjava 8 kriterijuma. **Trenutno stanje: 0/8 DoD kriterijuma ispunjeno**.[[3]](#fn3)

|  |  |  |
| --- | --- | --- |
| Kriterijum | Status | Razlog |
| 1. Funkcionalnost | ❌ FAIL | Zahtevi NISU implementirani (62.7% odstupanja od spec)[[2]](#fn2)[[4]](#fn4) |
| 2. API | ❌ FAIL | API NIJE potpisan ni dokumentovan (props nedostaju)[[2]](#fn2)[[4]](#fn4) |
| 3. Testovi | ❌ FAIL | Unit testovi NE PROLAZE (test errors)[[4]](#fn4) |
| 4. A11y | ❌ FAIL | Aria atributi NEDOSTAJU (ne ispunjava WCAG)[[2]](#fn2)[[3]](#fn3) |
| 5. Vizuelno | ❓ UNKNOWN | NIJE TESTRANO (no visual regression data)[[3]](#fn3) |
| 6. Tipovi i lint | ❌ FAIL | TypeScript IMA 78+ GREŠAKA[[4]](#fn4) |
| 7. Dokumentacija | ❓ UNKNOWN | Storybook stanje NIJE PROVERENO[[3]](#fn3) |
| 8. Bundle budžeti | ❓ UNKNOWN | NIJE PROVERENO[[3]](#fn3) |

**STATUS: Komponente NISU production-ready**[[3]](#fn3)

**Preporuke za implementaciju**

**Neposredne akcije (Sedmica 1)**

1. **STOP novih feature development** - Fokus na stabilizaciju i tech debt
2. **IMPLEMENTIRATI TypeScript strict mode kao blocker** - tsc --noEmit mora proći pre merge-a
3. **KREIRATI PR checklist prema DynUI-FS-03** - Link ka func spec obavezan; Props review prema spec dokumentaciji
4. **POSTAVITI Quality Gates u CI/CD**:
   * Gate A: TypeScript + ESLint (blocker)
   * Gate B: Unit tests (blocker)
   * Gate C: Axe accessibility (warning)[[4]](#fn4)[[3]](#fn3)

**Dugoročne mere**

1. **Kreirati Component Development Guide** - Pattern library sa examples; Props naming conventions; Type organization guidelines[[2]](#fn2)
2. **Uspostaviti Design System governance** - Weekly design review meetings; API change RFC process; Traceability matrix: spec → impl → test[[2]](#fn2)[[3]](#fn3)
3. **Implementirati automated documentation** - Auto-generate Props tables iz TypeScript; API diff checker između verzija; Storybook Controls sync sa Props interfejsima[[3]](#fn3)

**Zaključak**

Projekat DynUI ima solidnu funkcionalnu specifikaciju (DynUI-FS-01/02/03), ali **implementacija značajno odstupa od nje**. Glavni problem **NIJE u dizajnu sistema** već u:

* Nedoslednoj implementaciji specifikacije
* Odsustvu automatizovanih provera kvaliteta
* Nepostojanju code review procesa vezanog za spec

Predloženi plan rešavanja je **izvodljiv u okviru 6 sedmica**, ali zahteva:

✓ Punu posvećenost tima  
✓ Freeze na nove feature-e  
✓ Strict adherence ka specifikaciji  
✓ Implementaciju Quality Gates

**PREPORUKA**: ODOBRITI plan i startovati odmah kako bi se izbeglo dalje gomilanje tehničkog duga i kako bi komponente postale production-ready.[[4]](#fn4)[[3]](#fn3)

⁂

1. paste.txt

1. [DynUI-FS-02-Katalog-i-API.md](http://DynUI-FS-02-Katalog-i-API.md)

1. [DynUI-FS-03-Quality-Gates-CI-CD.md](http://DynUI-FS-03-Quality-Gates-CI-CD.md)

1. [DynUI-FS-01-Komponente-Audit-i-Zahtevi.md](http://DynUI-FS-01-Komponente-Audit-i-Zahtevi.md)