

Стандарт UI-спецификаций, который действительно работает 🚀

Прагматическая структура для передачи дизайна в разработку

Почему это важно 💡

Передачи дизайна часто проваливаются. Разработчики трижды пересобирают экраны. Продакт-менеджеры недоумевают, почему итоговые фичи не совпадают с макетами Figma. Эта структура решает проблему через **машиночитаемые спецификации**, которые компилируются напрямую в код платформы.

Ключевые выгоды:

- Внедрение быстрее на 50% благодаря детерминированным спецификациям
- Нулевая техническая задолженность по доступности через встроенное соответствие WCAG
- Автоматизированное тестирование от спецификации до продакшена

Основная архитектура 🏗️

Структура артефакта

YMTL—<Домен>—<Экран>—<Последовательность>
v1.2.3 | @владелец | Статус: Ready

Каждая спецификация содержит четыре слоя:

1. **Метаданные** — владельцы, версии, целевые платформы
2. **Композиция** — иерархия компонентов с токенами дизайна
3. **Поведение** — машины состояний и взаимодействия
4. **Валидация** — тестовые сценарии и крайние случаи

Масштабируемые токены дизайна 🎨

Используется формат W3C DTSG для обмена между инструментами:

```
{  
  "color.primary": "#007AFF",  
  "spacing.card": "16dp",  
  "radius.button": "8dp",  
  "elevation.modal": "24dp"  
}
```

Переопределения платформ сохраняют нативность. Android использует Material Design. iOS следует Human Interface Guidelines. Web адаптируется к ограничениям вьюпорта.

Управление состояниями

Стандартный словарь состояний

- **Default** → **Hover** → **Pressed** → **Success/Error**
- **Loading** — скелетоны
- **Empty** — с призывом к действию
- **Disabled** — с понятными путями восстановления

Шаблоны взаимодействия

Поведение задаётся декларативно:

```
on_click:  
  condition: user.authenticated  
  action: navigate("route:/dashboard")  
  analytics: track("button_clicked", {cta: "primary"})  
  animation: ease_out_cubic(300ms)
```

Поддерживаются сложные условия: `(feature.enabled && user.premium) || user.admin`

Доступность как базовое требование

Размеры для касаний

- **iOS**: минимум 44pt
- **Android**: базово 48dp
- **Web**: рекомендовано 44px

Соответствие WCAG 2.2

- Контраст **4.5:1** для текста
- Контраст **3:1** для интерактивных элементов
- Фокус не должен зависеть только от цвета
- Все интерактивные элементы имеют **ARIA-метки**

Масштабируемая интернационализация 🌐

ICU MessageFormat

```
{count, plural,  
  =0 {Нет элементов}  
  one {# элемент}  
  other {# элементов}  
}
```

Поддержка RTL

- Зеркальная вёрстка с позиционированием `start/end`
- +120% длины текста для романских языков
- +200% для немецких сложных слов

Интеграция QA 🧪

BDD-сценарии

Дано: у пользователя есть валидные учетные данные

Когда: нажата кнопка входа

Тогда: дашборд отображается за 2 секунды

Покрытие крайних случаев

Стандартные сценарии:

- **Сбой сети** → повтор с экспоненциальной задержкой
- **Пустое состояние** → иллюстрация и контекст
- **Частичные данные** → прогрессивный рендеринг
- **HTTP-ошибки** → дружелюбные сообщения

Архитектура аналитики

Таксономия событий

Именование `object_action` без лишних персональных данных:

- `button_clicked`
- `screen_viewed`
- `form_submitted`

Метрики производительности

- **Клиент:** FCP, LCP, TTI, CLS
- **Сервер:** задержки p50/p90/p95
- **Бизнес:** воронки конверсий, показатели вовлечённости

Спецификация API-контрактов

```
endpoint: /api/v2/users/{id}
method: GET
timeout: p95 < 500ms
retry: exponential_backoff(3)
cache: max-age=300
```

Схемы ссылаются на версионированные реестры для контрактного тестирования и обратной совместимости.

Методика внедрения

Стратегия

1. **Пилот** — новые фичи
2. **Валидация** — низкорисковые проекты
3. **Масштабирование** — после подтверждения эффективности
4. **Ретрофит** — постепенное внедрение в легаси

Интеграция инструментов

- **Figma** → экспорт UUISS через плагин

- **Android Studio** → генерация Compose-шаблонов
- **Xcode** → генерация SwiftUI-view
- **CI/CD** → автоматическая проверка соответствия

Модель управления

Три владельца:

- **Продукт** — метрики успеха
- **Дизайн** — пользовательский опыт
- **Разработка** — техническая реализуемость

Управление изменениями требует анализа влияния, увеличения версии и согласования.

Итог 🎯

Эта структура переводит UI-разработку из «ремесленного угадывания» в инженерную дисциплину. Команды выпускают быстрее, с меньшим числом багов и лучшей доступностью. Спецификация становится единым источником правды — больше никаких «что имел в виду дизайнер?».

Готовы внедрять? Начните с одной фичи. Измерьте эффект. Масштабируйте.

Полная спецификация со схемами и примерами находится в вашем репозитории дизайн-системы. Этот документ — практическое руководство по её применению.