



**Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

# **Метод динамического отображения изменений пользовательского интерфейса на основе обработки изменений XML-файла**

**Студент:** Егорова Полина Александровна, ИУ7-84Б

**Научный руководитель:** Никульшина Татьяна Александровна

# Критерии классификации методов отображения интерфейса

- **Адаптивность интерфейса:** возможность создания интерфейса, адаптируемого для экранов разного размера или двух ориентаций.
- **Интеграция в существующий код:** возможность постепенного внедрения метода в существующий код или совмещения с другими UI–фреймворками.
- **Механизм обработки изменений интерфейса:** наличие механизма, посредством которого отслеживаются обновления UI–компонента и инициируется его повторное отображение без участия разработчика.
- **Горячая перезагрузка:** возможность отображать изменения интерфейса, внесенные в код по время исполнения, без перекомпиляции всего приложения.

# Цель и задачи работы

**Целью** работы является разработка метода динамического отображения пользовательского интерфейса на основе обработки XML–файлов

## **Задачи:**

- выявить критерии, по которым можно классифицировать методы отображения пользовательского интерфейса;
- рассмотреть существующие методы отображения интерфейса и классифицировать их согласно выдвинутым критериям;
- разработать метод динамического отображения изменений интерфейса на основе обработки изменений XML–файла;
- разработать программное обеспечение, реализующее представленный метод;
- сравнить скорость внесения изменений в интерфейс разработанной и существующей реализаций.

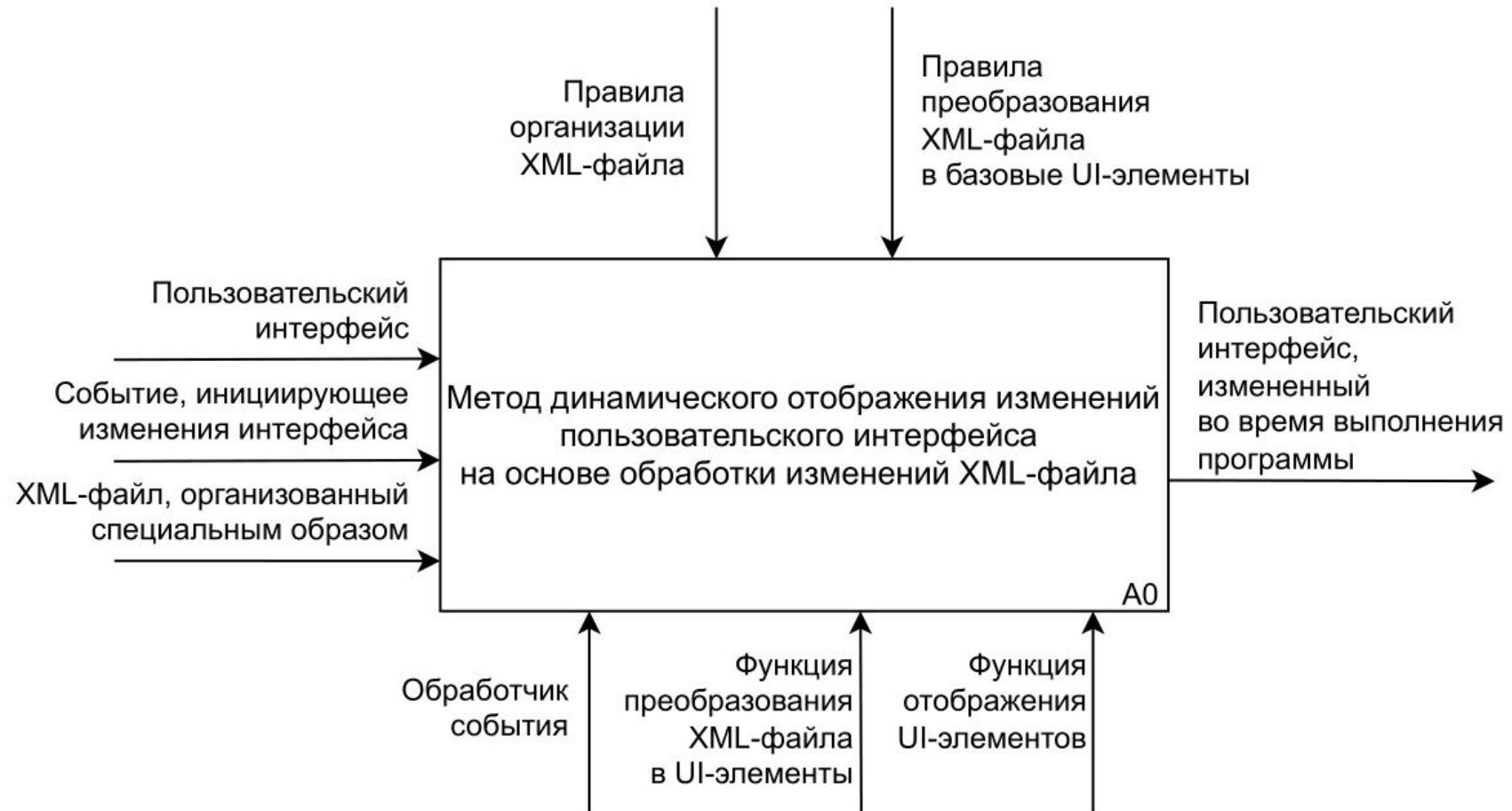
# Классификация методов отображения пользовательского интерфейса

Класс метода	Название	Адаптивность интерфейса	Интеграция в существующий код	Механизм обработки изменений интерфейса	Горячая перезагрузка
Frontend–разработка	React	+	+	+	+
	Vue	+	+	+	+
	Angular	+	+	+	+
Кроссплатформенная мобильная разработка	Flutter	+	+	+	+
	React Native	+	+	+	+
Нативная мобильная разработка Android	Метод с использованием XML	+	+	-	-
	Jetpack Compose	+	+	-	-
Нативная мобильная разработка iOS	UIKit	+	+	-	-
	SwiftUI	+	+	-	-

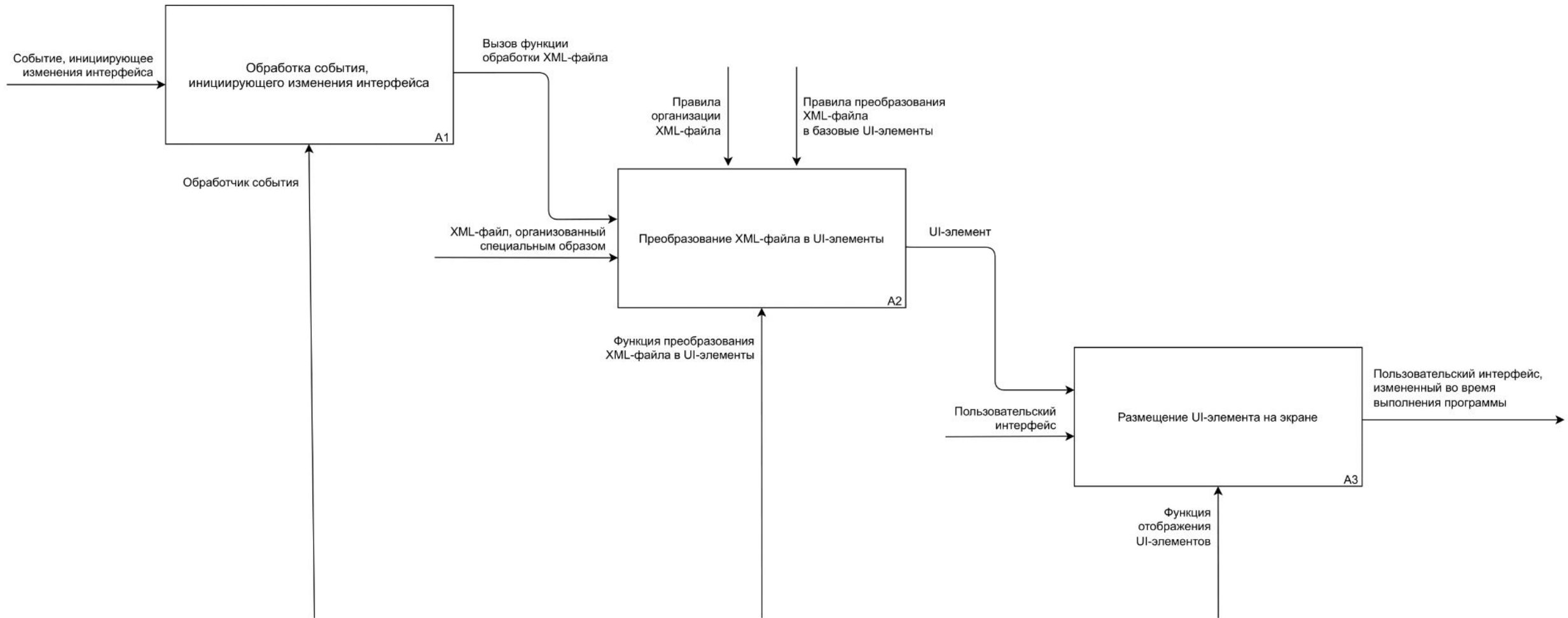
# Классификация методов отображения пользовательского интерфейса

Класс метода	Название	Адаптивность интерфейса	Интеграция в существующий код	Механизм обработки изменений интерфейса	Горячая перезагрузка
Frontend–разработка	React	+	+	+	+
	Vue	+	+	+	+
	Angular	+	+	+	+
Кроссплатформенная мобильная разработка	Flutter	+	+	+	+
	React Native	+	+	+	+
Нативная мобильная разработка Android	Метод с использованием XML	+	+	-	-
	Jetpack Compose	+	+	-	-
Нативная мобильная разработка iOS	UIKit	+	+	-	-
	SwiftUI	+	+	-	-

# Постановка задачи



# Разработанный метод



# Способ организации XML–файла

- Первый тег XML–файла содержит в себе название базового UI–элемента, его свойства и их значения.
- Название свойства и значение разделяет знак равенства «=».
- Значение свойства заключается в кавычки «“”».
- Второй тег содержит знак «\» и название базового элемента.
- Для создания вложенности элементов среди свойств элемента–родителя необходимо указать теги дочернего элемента.
- Разметка для дочернего элемента задается по правилам, аналогичным правилам создания родительского элемента.

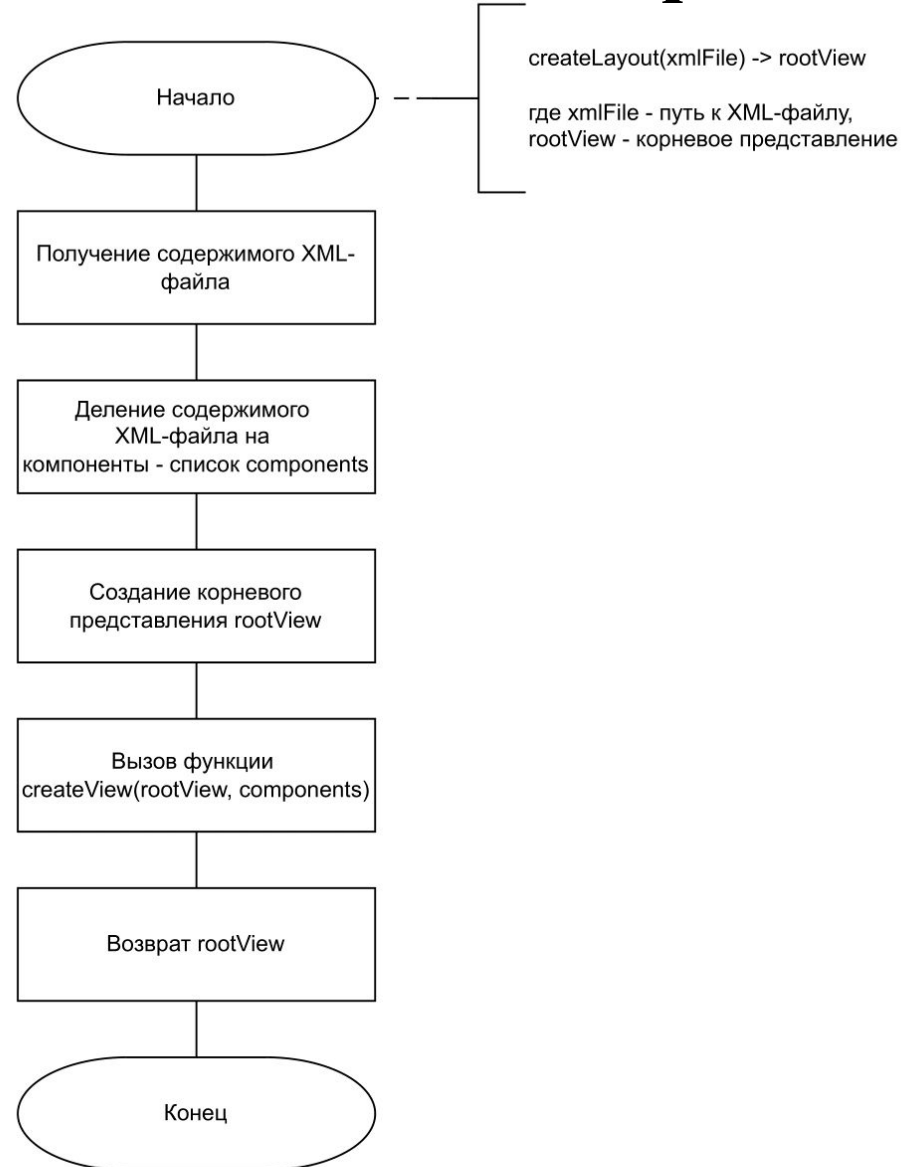
Пример:

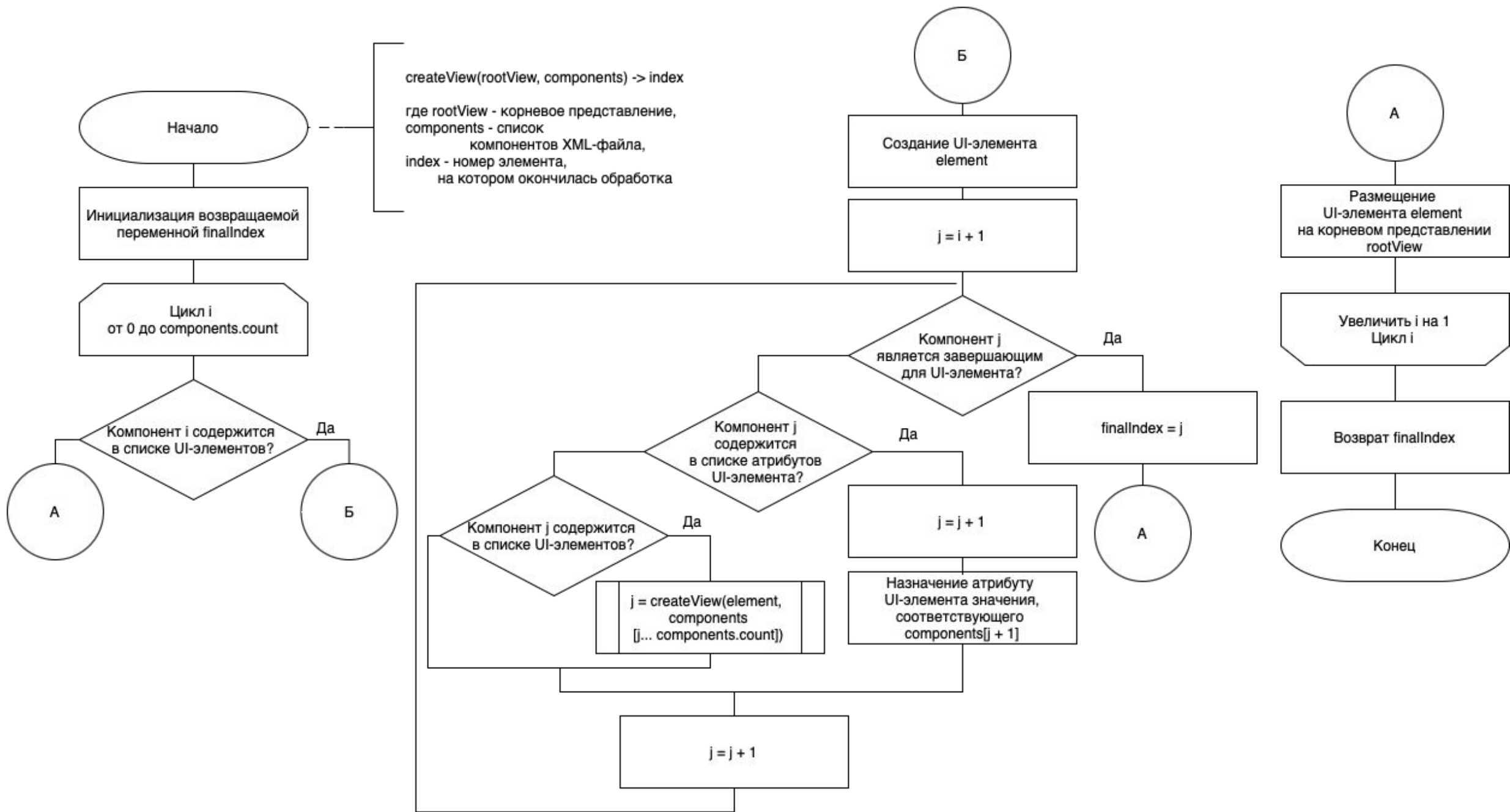
Разметка UILabel, занимающего 100% площади экрана и содержащего слово «Hello»

```
<UILabel  
  
        width="100%"  
        height="100%"  
        textAlignment="left"  
        text="Hello">  
  
</UILabel>
```



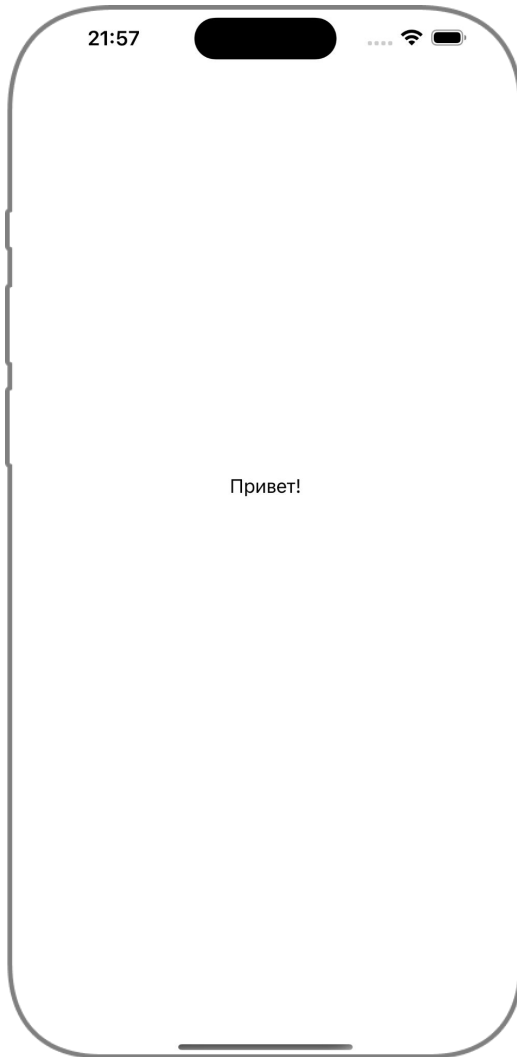
# Алгоритм работы функции создания представления, содержащего описанные в XML-файле элементы





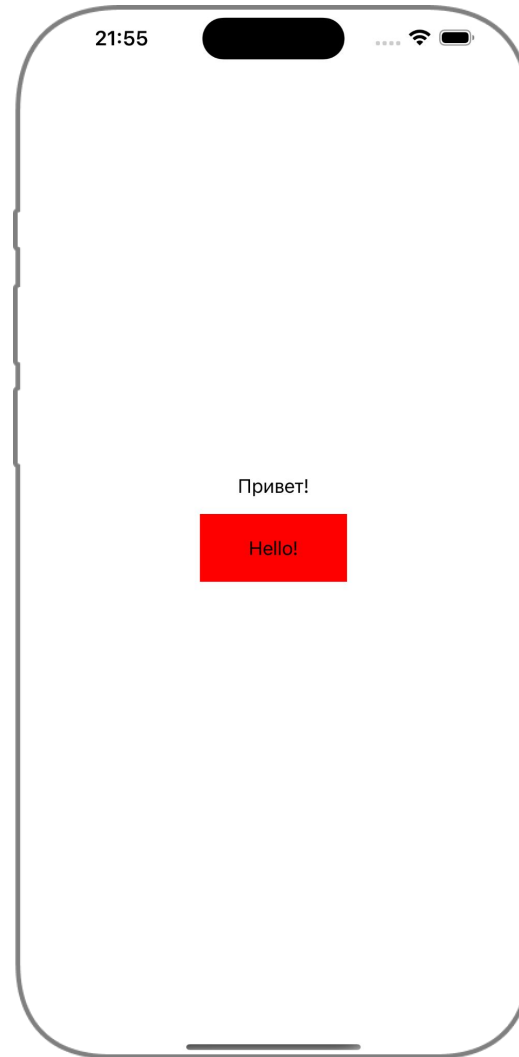
# Пример работы

До внесения изменений



```
<UIView
  width="100%"
  height="100%"
  backgroundColor="white">
  <UILabel
    width="100%"
    height="50"
    topAnchor="400"
    leftAnchor="0"
    text="Привет!"
    textAlignment="center">
  </UILabel>
</UIView>
```

После внесения изменений



```
<UIView
  width="100%"
  height="100%"
  backgroundColor="white">
  <UILabel
    width="100%"
    height="50"
    topAnchor="400"
    leftAnchor="0"
    text="Привет!"
    textAlignment="center">
  </UILabel>
  <UIView
    leftAnchor="150"
    rightAnchor="-150"
    height="60"
    topAnchor="450"
    backgroundColor="red">
    <UILabel
      leftAnchor="0"
      rightAnchor="0"
      bottomAnchor="0"
      topAnchor="0"
      text="Hello!"
      textAlignment="center">
    </UILabel>
  </UIView>
</UIView>
```

# Постановка исследования

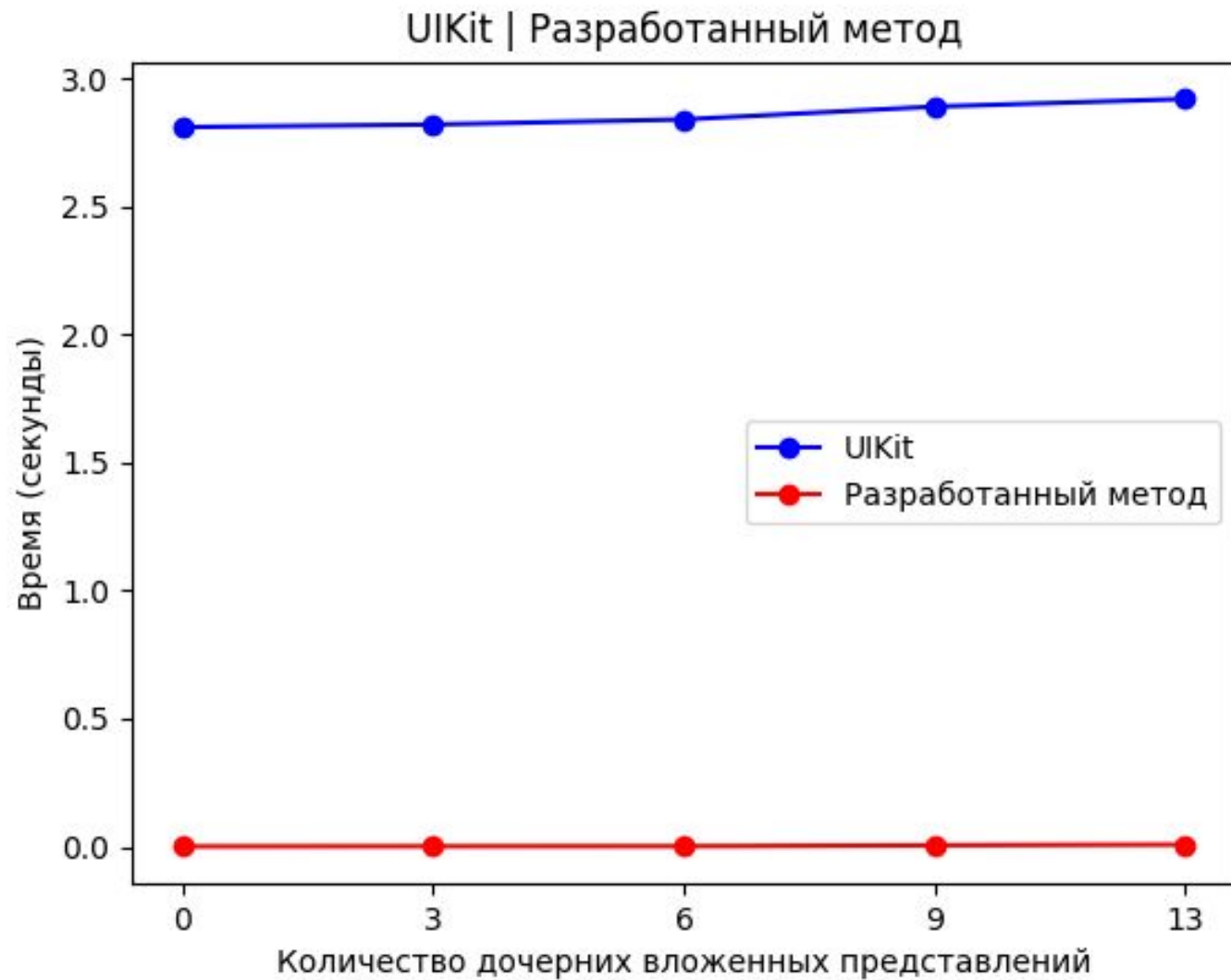
В рамках исследования была оценена эффективность реализованного метода путем сравнения скорости внесения изменений в интерфейс с существующей реализацией.



# Результаты исследования



# Результаты исследования



# Заключение

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель — разработан метод динамического отображения изменений пользовательского интерфейса путем обработки изменений XML–файла.

Были решены следующие задачи:

- выявлены критерии, по которым можно классифицировать методы отображения пользовательского интерфейса;
- классифицированы существующие методы отображения интерфейса согласно выдвинутым критериям;
- разработан метод динамического отображения изменений интерфейса на основе обработки изменений XML–файла;
- разработано программное обеспечение, реализующее представленный метод;
- произведено сравнение скорости внесения изменений в интерфейс разработанной и существующей реализаций.