Пользовательские интерфейсы в автоматизированных системах: проблемы разработки

**Человек:** Предметом исследования является процесс разработки программного обеспечения автоматизированных систем управления. Объект исследования – разработка пользовательских интерфейсов программных средств автоматизации управления. Общепризнанное перспективное направление повышения эффективности применения организационно-технических систем - автоматизация управления ими, обеспечивающая повышение оперативности и обоснованности принимаемых решений. Существенное влияние на эффективность любой автоматизированной системы оказывает её программное обеспечение. В первую очередь это относится к прикладному программному обеспечению. Разработка прикладных программ сопряжена с определёнными трудностями, в том числе, связанных с созданием пользовательских интерфейсов. Анализ практики разработки показывает наличие ряда проблем в данной области, определяющихся тем, что данная проблема находится на стыке научных дисциплин: теории управления, эргономики, технической эстетики и психологии. В ходе исследования использованы общенаучные методы анализа и синтеза. В обзорной статье проведён анализ факторов, влияющих на эффективность разработки пользовательских интерфейсов. На основе их анализа синтезированы предложения по решению проблемы, основанные на использовании средств стандартизации, унификации и прототипирования. Анализ показал, что для условий разработки прикладного программного обеспечения автоматизированных систем управления наибольшую эффективность обеспечивает последний из перечисленных подходов, а именно – использование специализированных систем прототипирования. Предложено доработать нормативную документацию, задающую разработку автоматизированных систем управления, для реализации в структуре процесса создания подобных систем обязательного этапа прототипирования пользовательских интерфейсов

**Key words:** прототипирование, макетирование, стандартизация и унификация, разработка интерфейсов, пользовательский интерфейс, автоматизация управления, разработка программного обеспечения, жизненный цикл программы, этап апробации, поддержка принятия решений

=================================

**FastText\_KMeans\_Clean:** В нашей стране эти требования устанавливаются ГОСТ серии 34 (для АСУ) и серии 19 (для отдельных программ), описываются в техническом задании на опытно-конструкторскую работу (ОКР) и детализируются в форме руководящих указаний главного конструктора ОКР. Впрочем, как показывает опыт применения таких средств, они лишь автоматизируют подход к использованию унифицированных компонентов, упрощая труд разработчиков, не внося в него ничего принципиально нового в части создания эффективных интерфейсных форм и не устраняя недостатки методов на основе стандартизации и унификации. Краткая характеристика. Позволяет преобразовывать прототипы в конечный продукт с помощью приложения Expression Suite. FormBuilder for Drupal. Профессиональный инструмент, для проектирования графических пользовательских интерфейсов (GUIs) прикладных программ и вебсайтов Обеспечивает создание интерфейсов, аннотаций к ним, построение раскадровок для определения рабочего прототипа.

**Key words part:** 0.5161290322580645

=================================

**FastText\_KMeans\_Raw/:** В то же время, анализ практики создания программного обеспечения показывает, что подходы, основанные на унификации компонентов, обеспечивают лишь частичное решение проблемы создания эффективного пользовательского интерфейса, в первую очередь – в части разработки компонентов общего и общесистемного программного обеспечения. Краткая характеристика. FormBuilder for Drupal. Профессиональный инструмент, для проектирования графических пользовательских интерфейсов (GUIs) прикладных программ и вебсайтов Обеспечивает создание интерфейсов, аннотаций к ним, построение раскадровок для определения рабочего прототипа.

**Key words part:** 0.6451612903225806

=================================

**FastText\_PageRank\_Clean/:** Вероятно, ещё большее количество случаев просто не предано огласке. Определяются они следующим. Есть ли выход из сложившейся ситуации?. Краткая характеристика. Виртуальный HTML-редактор. Expression Blend + SketchFlow. FormBuilder for Drupal. Justinmind Prototyper.

**Key words part:** 0.3225806451612903

=================================

**FastText\_PageRank\_Raw/:** Определяются они следующим. Времени на доработку остаётся крайне мало. Есть ли выход из сложившейся ситуации?. Краткая характеристика. Виртуальный HTML-редактор. Expression Blend + SketchFlow. FormBuilder for Drupal. Justinmind Prototyper.

**Key words part:** 0.3225806451612903

=================================

**Mixed\_ML\_TR/:** - слабое привлечение к процессу разработки специалистов смежных областей, способных оценить потребности пользователя и сформулировать пользовательские качества интерфейса уже на этапе разработки;. В то же время, анализ практики создания программного обеспечения показывает, что подходы, основанные на унификации компонентов, обеспечивают лишь частичное решение проблемы создания эффективного пользовательского интерфейса, в первую очередь – в части разработки компонентов общего и общесистемного программного обеспечения. В том числе в части разработки специализированных АСУ, например, ГОСТ серии РВ 1210. Для реализации принципа макетирования, как правило, используются штатные средства разработки программ. Таким образом, существующие подходы обеспечивают только частичную оптимизацию процесса разработки пользовательских интерфейсов и ограничены в использовании либо по функционалу, либо по времени. Простое средство разработки внешнего вида экранных форм. Приложение, позволяющее проектировать экранные формы web-приложений, экспортировать интерактивные HTML-прототипы и документацию. Во-первых , как показал анализ возможностей существующих методов и средств разработки пользовательских интерфейсов, подход с использованием прототипирования обладает преимуществами других методов, не повторяя от их недостатков. А сокращение времени – это практически всегда сокращение затрат.

**Key words part:** 0.7419354838709677

=================================

**MultiLingual\_KMeans/:** - слабое привлечение к процессу разработки специалистов смежных областей, способных оценить потребности пользователя и сформулировать пользовательские качества интерфейса уже на этапе разработки;. В том числе в части разработки специализированных АСУ, например, ГОСТ серии РВ 1210. Для реализации принципа макетирования, как правило, используются штатные средства разработки программ. Простое средство разработки внешнего вида экранных форм. Приложение, позволяющее проектировать экранные формы web-приложений, экспортировать интерактивные HTML-прототипы и документацию. А сокращение времени – это практически всегда сокращение затрат.

**Key words part:** 0.5806451612903226

=================================

**Multilingual\_PageRank/:** Вероятно, ещё большее количество случаев просто не предано огласке. Входящие в POSIX стандарты принято делить на четыре группы [8,9,10]:. Определяются они следующим. Времени на доработку остаётся крайне мало. Это существенно снижает эффективность данной методологии. Есть ли выход из сложившейся ситуации?. Краткая характеристика. А сокращение времени – это практически всегда сокращение затрат.

**Key words part:** 0.3225806451612903

=================================

**RuBERT\_KMeans\_Without\_ST/:** - определяющие взаимодействие в распределенных открытых системах и телекоммуникационных сетях, а также защиту информации (IEEE 1003.8, -12, -15,-17,-6);. В её рамках обеспечивается создание прототипов с минимальной функциональностью но с полным набором входных и выходных форм и совместная работа с ними будущего пользователя и разработчика с целью оперативного уточнения требований к интерфейсам программ. На практике, для прототипирования могут использоваться самые разные подходы и реализующие их средства: от простейших визуальных форм, описываемых в Power Point или Visio Professional до специализированных систем, обеспечивающих не только визуализацию, но и имитирующих реакцию интерфейсных форм на действия пользователя. Профессиональный HTML-редактор для создания статических или интерактивных опытных образцов программного обеспечения или вебсайта. При анализе данных таблицы, обращает на себя внимание существенное различие в сложности и функциональности систем прототипирования, а также то, что большинство из них используется для разработки web-интерфейсов.

**Key words part:** 0.5806451612903226

=================================

**RuBERT\_KMeans\_With\_ST/:** На практике, для прототипирования могут использоваться самые разные подходы и реализующие их средства: от простейших визуальных форм, описываемых в Power Point или Visio Professional до специализированных систем, обеспечивающих не только визуализацию, но и имитирующих реакцию интерфейсных форм на действия пользователя. Профессиональный инструмент, для проектирования графических пользовательских интерфейсов (GUIs) прикладных программ и вебсайтов Обеспечивает создание интерфейсов, аннотаций к ним, построение раскадровок для определения рабочего прототипа. Web-инструмент для прототипирования, с возможностью совместной работы в стандартном режиме и режиме Sketch. Во-первых , как показал анализ возможностей существующих методов и средств разработки пользовательских интерфейсов, подход с использованием прототипирования обладает преимуществами других методов, не повторяя от их недостатков.

**Key words part:** 0.5483870967741935

=================================

**RUBERT\_page\_rank\_Without\_ST/:** При общепризнанной важности первых, вторые, как показывает практика, ничуть не менее важны: именно от них зависит внедрение результатов теоретических исследований в практику, обеспечивающее поступательное развитие технологий. Входящие в POSIX стандарты принято делить на четыре группы [8,9,10]:. При средней продолжительности ОКР по разработке программной продукции в 2-3 года, первая апробация, как правило, проводится не ранее, чем через год. Краткая характеристика. Приложение, позволяющее проектировать экранные формы web-приложений, экспортировать интерактивные HTML-прототипы и документацию.

**Key words part:** 0.4516129032258064

=================================

**RUBERT\_page\_rank\_With\_ST/:** Определяются они следующим. Времени на доработку остаётся крайне мало. Краткая характеристика. FormBuilder for Drupal. Justinmind Prototyper.

**Key words part:** 0.3225806451612903

=================================

**RUSBERT\_KMeans\_Without\_ST/:** К организационным подходам можно отнести формирование общих требований к интерфейсам программ их доведение до разработчиков и контроль исполнения. Инструмент, обеспечивающий создание экранных форм web-приложений с возможностью определения их поведения через описание UseCase-диаграмм. Впрочем, гораздо важнее не проблема сертификации, а то, что использование прототипирования не определено нормативными документами, определяющими разработку программной продукции. Соответственно, потребуются средства оптимизации их разработки.

**Key words part:** 0.5483870967741935

=================================

**RUSBERT\_KMeans\_With\_ST/:** В настоящее время именно он является основой приведения функционала и интерфейсов прикладных программ в соответствие требованиям заказчика, их доработки под требования пользователей. Но, как показывает личный опыт автора, в применении методологии макетирования имеются некоторые проблемы. Web-инструмент для прототипирования, с возможностью совместной работы в стандартном режиме и режиме Sketch. Впрочем, гораздо важнее не проблема сертификации, а то, что использование прототипирования не определено нормативными документами, определяющими разработку программной продукции.

**Key words part:** 0.5806451612903226

=================================

**RUSBERT\_page\_rank\_Without\_ST/:** Вероятно, ещё большее количество случаев просто не предано огласке. Определяются они следующим. Времени на доработку остаётся крайне мало. Expression Blend + SketchFlow. Justinmind Prototyper.

**Key words part:** 0.3225806451612903

=================================

**RUSBERT\_page\_rank\_With\_ST/:** Определяются они следующим. Времени на доработку остаётся крайне мало. Есть ли выход из сложившейся ситуации?. Краткая характеристика. Expression Blend + SketchFlow.

**Key words part:** 0.3225806451612903

=================================

**Simple\_PageRank/:** Порядок применения этих стандартов регулируется Руководством POSIX Guide IEEE 1003.0. С использованием указанного набора стандартов, руководителем проекта формируются правила описания интерфейсов, обязательные для исполнения всеми участниками процесса [11,12,13,14]. В её рамках обеспечивается создание прототипов с минимальной функциональностью но с полным набором входных и выходных форм и совместная работа с ними будущего пользователя и разработчика с целью оперативного уточнения требований к интерфейсам программ. Но главное, не наличие и функционал средств, а сам факт использования метода прототипирования, применения технологии, которая позволяет существенно сократить цикл взаимодействия между пользователем и разработчиком. Можно возразить, что использование методов прототипирования может нарушать 188-ФЗ от 29.07.2015 года "О внесении изменений в Федеральный закон "Об информации, информационных технологиях и о защите информации" и статью 14 Федерального закона "О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд", так как большинство программных средств, реализующих этот подход, не включены в реестр доверенного отечественного программного обеспечения. В обозримой перспективе, вполне вероятно, потребуется создавать не только стандартные графические, но и 3D-интерфейсы, а также аудио- и нейро-интерфейсные компоненты. Нельзя забывать, что для использования средств прототипирования потребуется корректировка нормативных документов, регламентирующих ведение ОКР по разработке программной продукции АСУ: как в части введения этапа прототипирования, так и привлечения специалистов из смежных областей.

**Key words part:** 0.6451612903225806

=================================

**TextRank/:** В то же время, анализ практики создания программного обеспечения показывает, что подходы, основанные на унификации компонентов, обеспечивают лишь частичное решение проблемы создания эффективного пользовательского интерфейса, в первую очередь – в части разработки компонентов общего и общесистемного программного обеспечения. Таким образом, существующие подходы обеспечивают только частичную оптимизацию процесса разработки пользовательских интерфейсов и ограничены в использовании либо по функционалу, либо по времени. В её рамках обеспечивается создание прототипов с минимальной функциональностью но с полным набором входных и выходных форм и совместная работа с ними будущего пользователя и разработчика с целью оперативного уточнения требований к интерфейсам программ. Профессиональный инструмент, для проектирования графических пользовательских интерфейсов (GUIs) прикладных программ и вебсайтов Обеспечивает создание интерфейсов, аннотаций к ним, построение раскадровок для определения рабочего прототипа. Как показал опыт работы автора над разработкой АСУ, при использовании средств прототипирования, длительность цикла взаимодействия при разработке пользовательского интерфейса может быть сокращена до 3-5 месяцев. Во-первых , как показал анализ возможностей существующих методов и средств разработки пользовательских интерфейсов, подход с использованием прототипирования обладает преимуществами других методов, не повторяя от их недостатков.

**Key words part:** 0.7096774193548387

=================================

**TF-IDF\_KMeans/:** В ней, как и в процессе создания любого наукоёмкого продукта, имеется чисто научная (фундаментальная, методологическая) и научно-практическая (технологическая) составляющая. 2.Существующие подходы к решению проблем разработки пользовательских интерфейсов. В нашей стране эти требования устанавливаются ГОСТ серии 34 (для АСУ) и серии 19 (для отдельных программ), описываются в техническом задании на опытно-конструкторскую работу (ОКР) и детализируются в форме руководящих указаний главного конструктора ОКР. Профессиональное web-приложение для создания интерактивных экранных форм.

**Key words part:** 0.5483870967741935

=================================

**Текст:** История создания и применения автоматизированных систем управления (АСУ) насчитывает уже несколько десятилетий. В ней, как и в процессе создания любого наукоёмкого продукта, имеется чисто научная (фундаментальная, методологическая) и научно-практическая (технологическая) составляющая. Научно-исследовательские и опытно-конструкторский работы, выполняемые в целях создания наукоёмкой продукции, принято разделять на работы «первого» (фундаментальные исследования) и «второго» (прикладные) рода. При общепризнанной важности первых, вторые, как показывает практика, ничуть не менее важны: именно от них зависит внедрение результатов теоретических исследований в практику, обеспечивающее поступательное развитие технологий.. Одна из практических научных задач «второго рода» – оптимизация разработки программного обеспечения автоматизированных систем управления, обеспечивающих поддержку принятия решений, и в настоящее время остаётся одной из не до конца решенных научно-практических задач [1,2,3,4]. В том числе, а может быть в первую очередь – из-за проблем с организацией разработки интерфейсной части программ, реализуемых в автоматических системах поддержки принятия решений (СППР).. . 1.Некоторые проблемы создания пользовательских интерфейсов. Как показывает практика эксплуатации программного обеспечения, именно пользовательский интерфейс является «лицом» программы, определяющим удобство её использования и, как результат, желание пользователя её эксплуатировать в процессе выработки управленческих решений. Известно достаточное количество случаев, когда именно некачественный интерфейс мешал внедрению хорошей программы [5,6,7]. Вероятно, ещё большее количество случаев просто не предано огласке. Сложность указанной задачи определяется тем, что проблема разработки интерфейса является комплексной, находящейся на стыке дисциплин: математики, технической эстетики, психологии, эргономики. И, несмотря на достаточно долгую историю вопроса, рациональная организация разработки интерфейсов программ до настоящего времени не реализована в полном объёме и проблема её решения остаётся актуальной.. Как показывает опыт разработки программ, наличие указанной проблемы определяется целым рядом факторов:. - недостаточная унификацию интерфейсов, разрабатываемых для программных продуктов различного назначения, для разных операционных систем и технических платформ;. - слабое привлечение к процессу разработки специалистов смежных областей, способных оценить потребности пользователя и сформулировать пользовательские качества интерфейса уже на этапе разработки;. - не всегда рациональную организацию процесса разработки программной продукции, в первую очередь – в части организации взаимодействия с потребителем.. В совокупности, эти факторы не позволяют обеспечить требуемое качество интерфейсов разрабатываемых прикладных программ, чем определяется общее снижение эффективности автоматизированной поддержки принятия решений.. . 2.Существующие подходы к решению проблем разработки пользовательских интерфейсов. . Как показывает практика, чаще всего для оптимизации разработки интерфейсов используется наиболее простой и очевидный подход – унификация и стандартизация управляющих и отображающих элементов. Для их функциональной и визуальной унификации может быть использовано несколько подходов: как организационных, так и технологических.. К организационным подходам можно отнести формирование общих требований к интерфейсам программ их доведение до разработчиков и контроль исполнения. В нашей стране эти требования устанавливаются ГОСТ серии 34 (для АСУ) и серии 19 (для отдельных программ), описываются в техническом задании на опытно-конструкторскую работу (ОКР) и детализируются в форме руководящих указаний главного конструктора ОКР.. При формировании этих документов используются специализированные стандарты, созданные на основе международных аналогов, принятых к использованию РФ:. 1)В части функциональных требований, например, к отдельным компонентам двумерных графических интерфейсов WIMP (window, icon, menu, pointing device), в стандартах ISO 9241-12, ISO 9241-14, ISO 9241-16, ISO/IEC 10741, ISO/IEC 12581 и других.. 2)В части эргономических характеристик, в международных стандартах ISO 9241-10, ISO/IEC 13407, а также ГОСТ Р ИСО/МЭК 12119-2000, ГОСТ Р ИСО/МЭК 9126-93 и ряде других.. Зарубежные разработчики, преимущественно, пользуются комплексом стандартов по организации интерфейсов переносимых операционных систем (Portable operating system interfaces — POSIX). Входящие в POSIX стандарты принято делить на четыре группы [8,9,10]:. - базовые, определяющие общие принципы построения, директивы, основы реализации и тестирования интерфейсов переносимых приложений, серии IEEE 1003.1, -2, -3, -4, -7;. - конкретизирующие интерфейсы для операционных платформ, серии IEEE 1003.5, -9, -16, -19, -20;. - определяющие взаимодействие в распределенных открытых системах и телекоммуникационных сетях, а также защиту информации (IEEE 1003.8, -12, -15,-17,-6);. - регламентирующие процесс разработки программ, серия IEEE 1003.10, -11, -13, -14, -18.. Порядок применения этих стандартов регулируется Руководством POSIX Guide IEEE 1003.0. С использованием указанного набора стандартов, руководителем проекта формируются правила описания интерфейсов, обязательные для исполнения всеми участниками процесса [11,12,13,14].. Технологические подходы обеспечиваются применением специализированных программных средств и библиотек интерфейсных компонентов, таких, как Linux Mint, UI-kit и им подобных. Они обеспечивают выбор перечня и типов интерфейсных компонентов, а также управление их списками, в том числе централизованное. Впрочем, как показывает опыт применения таких средств, они лишь автоматизируют подход к использованию унифицированных компонентов, упрощая труд разработчиков, не внося в него ничего принципиально нового в части создания эффективных интерфейсных форм и не устраняя недостатки методов на основе стандартизации и унификации.. В целом подходы, ориентированные на унификацию, упрощают работу по созданию пользовательских интерфейсов в рамках каждого отдельного проекта. В то же время, анализ практики создания программного обеспечения показывает, что подходы, основанные на унификации компонентов, обеспечивают лишь частичное решение проблемы создания эффективного пользовательского интерфейса, в первую очередь – в части разработки компонентов общего и общесистемного программного обеспечения. Проблемы создания интерфейсов прикладных программ их применением решаются в ограниченном объёме.. Практика разработки АСУ показывает, что возможен ещё один подход к повышению качества разрабатываемых пользовательских интерфейсов – макетирование программ. В любой из известных методологий разработки программ: рациональное программирование RUP (Rational Unified Process), итеративно-инкрементальный метод OpenUP, технологии быстрой разработки приложений RAD (rapid application development), методология MSF (Microsoft Solutions Framework), методология разработки программ с сертифицируемым уровнем надёжности Cleanroom (Cleanroom Software Engineering), группа технологий гибкого программирования Agile, в том или ином виде присутствует этап апробации программ [15,16,17,18,19]. В настоящее время именно он является основой приведения функционала и интерфейсов прикладных программ в соответствие требованиям заказчика, их доработки под требования пользователей. Более того, данный этап, хоть не обязательный, но утверждён ГОСТ по разработке программной продукции. В том числе в части разработки специализированных АСУ, например, ГОСТ серии РВ 1210.. Методология макетирования, реализуемая через этап апробации, подразумевает создание пробной версии полнофункционального продукта c реализацией базовой части функционала (так называемой ß-версии) и её проверки на пригодность пользователями. Для реализации принципа макетирования, как правило, используются штатные средства разработки программ.. Но, как показывает личный опыт автора, в применении методологии макетирования имеются некоторые проблемы. Определяются они следующим. В структуре технического задания на ОКР по разработке программной продукции, описание интерфейсов программ нормативными и руководящими документами не предусмотрено. Первоначальный вариант интерфейса программист разрабатывает сам, с учётом своего видения проблемы. При средней продолжительности ОКР по разработке программной продукции в 2-3 года, первая апробация, как правило, проводится не ранее, чем через год. С учётом времени проведения самой апробации и срока обработки её результатов, до программиста требования пользователя по уточнению интерфейсов доходят через 13-15 месяцев. Времени на доработку остаётся крайне мало. Это существенно снижает эффективность данной методологии.. Есть ещё один вариант частного решения данной проблемы – создание интерфейсов, адаптируемых под пользователя (настраиваемые интерфейсы) [20,21,22]. Но даже самые совершенные адаптивные интерфейсы настраиваются в определённых границах, которые тоже требуется задавать при разработке, что не слишком упрощает процесс создания входных и выходных форм программ.. Таким образом, существующие подходы обеспечивают только частичную оптимизацию процесса разработки пользовательских интерфейсов и ограничены в использовании либо по функционалу, либо по времени. Есть ли выход из сложившейся ситуации?. . 3.Прототипирование программ как метод совершенствования разработки пользовательских интерфейсов. . Как показывает анализ мирового опыта разработки программной продукции, альтернативой макетированию может стать методология быстрого прототипирования (RP-технологии: Rapid Prototyping). В её рамках обеспечивается создание прототипов с минимальной функциональностью но с полным набором входных и выходных форм и совместная работа с ними будущего пользователя и разработчика с целью оперативного уточнения требований к интерфейсам программ. В нашей стране прототипирование используется не так активно, как за рубежом, преимущественно – при создании web-приложений [23,24,25]. Что, впрочем, не отрицает перспектив использования данного подхода при разработке интерфейсов прикладных программ различного назначения.. На практике, для прототипирования могут использоваться самые разные подходы и реализующие их средства: от простейших визуальных форм, описываемых в Power Point или Visio Professional до специализированных систем, обеспечивающих не только визуализацию, но и имитирующих реакцию интерфейсных форм на действия пользователя. Общий обзор средств прототипирования по данным из работ [26,27], приводится в таблице.. . Таблица – Перечень некоторых программных средств прототипирования. . Наименование. Краткая характеристика. Powerpoint. Простое средство разработки внешнего вида экранных форм. Прототипирование с его использованием обеспечивается возможностью создания интерактивных «горячих» точек, которые можно использовать для переходов между слайдами и наличием эффектов анимации. Visio Professional. Простой редактор для описания внешнего вида экранных форм. Обеспечивает создание простейших прототипов с минимальной интерактивностью. MockupScreens. Несложный графический редактор для создания экранных форм без интерактивных возможностей. Обеспечивает распределённую работу в интерактивном режиме. Keynote. Несложный инструмент, обеспечивающий описание внешнего вида форм. Напоминает Power Point, при этом обеспечивает создание «кликабельных» объектов, открытие внешних ссылок, запуск видеофайлов. Flex. WYSIWYG редактор с поддержкой импорта визуальных форм из Adobe Illustrator. Имеется возможность экспорта Flash или Air приложений. Axure RP Pro. Инструмент, ориентированный на создание прототипов web-сайтов. Генерирует «кликабельные» HTML страницы и документы. Balsamiq Mockups. Программа, позвояющая быстро создавать макеты пользовательских форм интерфейсов web-сайтов и прикладных программ, сохранять их в виде XML-файлов. CogTool. Инструмент с открытым кодом, обеспечивающий разработку простых макетов пользовательского интерфейса. Coutline. Web-приложение для создания и просмотра интерактивных прототипов. Включает в себя функции по управлению проектом. Dreamweaver. Виртуальный HTML-редактор. Обеспечивает формирование и размещение элементов дизайна с помощью drag-and-drop, добавление элементов интерактивности, и редактирование кода для прототипирования пользовательских форм web-приложений. EasyPrototype. Приложение, позволяющее проектировать экранные формы web-приложений, экспортировать интерактивные HTML-прототипы и документацию. Expression Blend. WYSIWYG-инструмент для проектирования пользовательского интерфейса приложений, создаваемых на основе Windows Presentation Foundation. Обеспечивает разработку дизайна web-интерфейсов и графических настольных приложений. Может генерировать прототипы для Silverlight и WPF-приложений с интерактивными возможностями. Expression Blend + SketchFlow. Инструмент динамичного прототипирования, обеспечивающий создание концепций интерфейсов, которые выглядят как продукты Sketch. Позволяет преобразовывать прототипы в конечный продукт с помощью приложения Expression Suite. Expression Design. Профессиональный инструмент для графического дизайна и подготовки иллюстраций, который позволяет создавать элементы пользовательского интерфейса для настольных и web-приложений. Обеспечивает создание прототипов HTML, Silverlight и WPF-приложений с ограниченной интерактивностью. Fireworks. Растровый и векторный графический редактор для web-дизайнеров и разработчиков, позволяющий быстро создавать, редактировать и оптимизировать эскизы сайтов. Обеспечивает создание сложных интерактивных прототипов. Содержит библиотеку готовых настроек, хорошо интегрирован с продуктами Adobe Photoshop, Adobe Illustrator, Adobe Dreamweaver и Adobe Flash. Имеется возможность экспорта результатов в PDF или HTML. FlairBuilder. HTML-редактор, позволяющий создавать интерактивные экранные формы с помощью десктопного Air приложения. Формирует результат для просмотра в виде самостоятельного приложения. ForeUI. Профессиональный HTML-редактор для создания статических или интерактивных опытных образцов программного обеспечения или вебсайта. Обеспечивает создание макетов, моделирующих поведение приложения в браузере, в том числе в режиме распределённой работы. FormBuilder for Drupal. Профессиональный инструмент разработки интерактивных пользовательских форм. Имеет web-интерфейс с возможностью «перетаскивания» элементов на страницу. Создаёт работающие прототипы форм, в том числе в части выполнения требований к вводимым параметрам. GUI Design Studio. Профессиональный инструмент, для проектирования графических пользовательских интерфейсов (GUIs) прикладных программ и вебсайтов Обеспечивает создание интерфейсов, аннотаций к ним, построение раскадровок для определения рабочего прототипа. iPlotz. Профессиональное web-приложение для создания интерактивных экранных форм. Включает в себя базовый набор функций для управления проектом и обеспечения распределённой работы. iRise. Комплексный инструмент для моделирования бизнес-процессов с возможностью проектирования интерфейсов прикладных программ. Justinmind Prototyper. Инструмент, обеспечивающий создание экранных форм web-приложений с возможностью определения их поведения через описание UseCase-диаграмм. Позволяет экспортировать прототипы в формате HTML, генерировать текстовую документацию. JustProto. Web-приложение, ориентированное на удалённую работу, в том числе в реальном масштабе времени. Имеет развитые библиотеки типовых интерфейсных элементов. Pencil Project. Плагин для FireFox с функцией разработки прототипов. Схемы сайтов создаются непосредственно в браузере, в том числе в распределённом режиме. Обеспечивает создание активных экранных форм. Pidoco. Web-инструмент для прототипирования, с возможностью совместной работы в стандартном режиме и режиме Sketch. Интуитивно понятный drag-n-drop интерфейс. Protoscript. Простой скриптовой язык, обеспечивающий возможность добавления динамических элементов к существующим HTML-страницам. Protoshare. Web-приложение, ориентированное на групповую работу с интерактивными экранными формами. Screen Architect. Дополнение для инструмента UML-моделирования Enterprise Architect, обеспечивающее создание несложных прототипов интерфейсов. . При анализе данных таблицы, обращает на себя внимание существенное различие в сложности и функциональности систем прототипирования, а также то, что большинство из них используется для разработки web-интерфейсов. Последнее вполне объяснимо в рамках тенденции [28,29] активного развития UX/UI дизайна (User Experience/User Interface). В любом случае, каждый из перечисленных в таблице инструментов имеет свои особенности, преимущества и недостатки.. А наличие широкого спектра возможностей, от простейшего визуального PowerPoint до полнофункциональной GUI Machine Project (рисунок 1), обеспечивает, как показывает практика, то, что каждое из указанных средств находит своего пользователя. Использование того или иного из них определяется конкретными условиями разработки.. . . Рис.1.Форма работы с проектами GUI Machine Project. . Но главное, не наличие и функционал средств, а сам факт использования метода прототипирования, применения технологии, которая позволяет существенно сократить цикл взаимодействия между пользователем и разработчиком. Как показал опыт работы автора над разработкой АСУ, при использовании средств прототипирования, длительность цикла взаимодействия при разработке пользовательского интерфейса может быть сокращена до 3-5 месяцев. И это, при обеспечении приемлемой функциональности, позволяющей оценить не только визуальные, но и функциональные особенности интерфейсов. То есть при функциональности как при использовании метода апробации, обеспечивается сокращение времени «отклика» в 5-6 раз.. . Заключение. Таким образом, анализ ситуации, сложившейся с разработкой интерфейсов прикладных программ позволяет сделать ряд выводов.. Во-первых , как показал анализ возможностей существующих методов и средств разработки пользовательских интерфейсов, подход с использованием прототипирования обладает преимуществами других методов, не повторяя от их недостатков.. Во-вторых , использование метода прототипирования обеспечивает проведение до начала этапа апробации 3-4-х циклов согласования интерфейсных форм программ, что позволит на этом этапе работать уже с функционалом программ, а не тратить время на совершенствование их входных и выходных форм. А сокращение времени – это практически всегда сокращение затрат.. Более того , простота и сетевые возможности систем прототипирования позволят активно включить пользователя, других необходимых специалистов, непосредственно в разработку интерфейсных форм, перейдя от итерационного подхода к непрерывному процессу работы.. Можно возразить, что использование методов прототипирования может нарушать 188-ФЗ от 29.07.2015 года «О внесении изменений в Федеральный закон "Об информации, информационных технологиях и о защите информации" и статью 14 Федерального закона "О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд», так как большинство программных средств, реализующих этот подход, не включены в реестр доверенного отечественного программного обеспечения. В то же время, разместить необходимые средства разработки в реестре не так уж сложно, порядок включения достаточно прозрачный и подробно описан в законе. Ситуацию упрощает то, что многие средства прототипирования технически не связаны с кодом конечного продукта, чем снижается проблема информационной безопасности.. Впрочем, гораздо важнее не проблема сертификации, а то, что использование прототипирования не определено нормативными документами, определяющими разработку программной продукции.. В третьих , тенденции развития АСУ показывает, что функционал существующих средств прототипирования, как и других средств, используемых сейчас для обеспечения разработки интерфейсов, недостаточен. В обозримой перспективе, вполне вероятно, потребуется создавать не только стандартные графические, но и 3D-интерфейсы, а также аудио- и нейро-интерфейсные компоненты. Соответственно, потребуются средства оптимизации их разработки.. И последнее. Нельзя забывать, что для использования средств прототипирования потребуется корректировка нормативных документов, регламентирующих ведение ОКР по разработке программной продукции АСУ: как в части введения этапа прототипирования, так и привлечения специалистов из смежных областей. Но эти меры не слишком сложны и затратны, а их реализация обещает дать существенный прирост эффективности разработки прикладных программ.. Обобщая вышесказанное, можно сделать вывод, что использование методов прототипирования обеспечит оптимизацию цикла разработки пользовательских интерфейсов и, в итоге, повышение эффективности процесса разработки АСУ и СППР в целом.