

УДК 658.012.011.56

РАЗРАБОТКА ОБОБЩЕННОЙ МОДЕЛИ СОЗДАНИЯ ПРОГРАММНОГО КОМПЛЕКСА АВТОМАТИЗИРОВАННЫХ СИСТЕМ

В.М. Левыкин, М.А. Жигалов.

Харьковский национальный университет радиоэлектроники.

Была решена и обоснована задача создания обобщенной модели процесса разработки программного комплекса АС. В рамках данной задачи был проведен анализ существующих моделей разработки программного комплекса, на основании результатов которого были выявлены их преимущества и недостатки, а также была предложена новая обобщенная модель. Для разработанной модели был предложен алгоритм создания программного комплекса автоматизированных систем, применение которого позволяет дать оценку длительности, трудоемкости и других параметров данного процесса и облегчить выбор методологии проектирования. Для предложенного алгоритма была разработана математическая интерпретация при помощи метода регулярного выражения алгоритмов.

Ключевые слова: программный комплекс, модель создания, алгоритм разработки, регулярное выражение алгоритмов.

Постановка проблемы в общем виде

В рамках этой работы решается задача разработки обобщенной модели и алгоритма процесса создания программного комплекса автоматизированных систем. На текущий момент существует достаточно большое количество подобных моделей, и одной из проблем, рассматриваемых в статье, является разработка модели, которая учитывала бы основные аспекты процесса создания программного комплекса (ПК).

С развитием IT-технологий и высокой динамикой изменения ситуации на рынке становятся очень жесткими требования к функциям, выполняемым ПК, к процессу создания ИС, требования ко времени разработки отдельных приложений и системы в целом. Поэтому процессы разработки ПК требуют постоянного усовершенствования.

Впервые концепция жизненного цикла разработки ПК (Software life cycle, SLC) была предложена в 1970 г. Модель описывала последовательность событий, происходящих при разработке ПК. В последствии она претерпела множество изменений и доработок.

Под моделью процесса создания программного комплекса понимаем общее абстрактное представление данного процесса. Каждая такая модель представляет процесс создания ПК в каком-то своем "разрезе", используя только определенную часть всей информации о процессе.

Анализ последних исследований и публикаций

Специалисты считают, что на сегодняшний день существует настолько много различных типов проектов, что в рамках одномерной модели SLC они вряд ли могут быть описаны [1]. Современная точка зрения на данную проблему заключается в том, что необходимо использовать уникальные модели или комбинации моделей для уникальных проектов.

Среди существующих моделей можно выделить ряд наиболее используемых: каскадная, эволюционная, формальной разработки систем, а также модель разработки ПК на основе ранее созданных компонентов. Каскадная модель отображает основные базовые виды деятельности, выполняемые в процессе создания ПК, которые представляются как отдельные этапы этого процесса.

При использовании эволюционной модели разработки ПК последовательно перемежаются этапы формирования требований, разработки программного обеспечения (ПО) и его аттестации. На практике первоначально программная система быстро разрабатывается на основе некоторых абстрактных общих требований. Затем они уточняются и детализируются в соответствии с требованиями заказчика. Далее система дорабатывается и аттестуется в соответствии с новыми уточненными требованиями.

Модель формальной разработки ПК основана на разработке формальной математической спецификации программной системы и преобразовании этой спецификации посредством специальных математических методов в исполняемые программы. Проверка соответствия спецификации и системных компонентов также выполняется математическими методами.

Модель разработки ПК на основе ранее созданных компонентов предполагает, что отдельные составные части программной системы уже существуют. В этом случае реализация технологического процесса создания ПК подразумевает интеграцию отдельных компонентов в общее целое, а не их создание.

Каскадная и эволюционная модели разработки не очень широко используются на практике. Модель формальной разработки систем успешно применялась во многих проектах, но количество организаций-разработчиков, постоянно использующих этот метод, невелико. Использование готовых системных компонентов практикуется повсеместно, но большинство организаций не придерживаются в точности модели разработки ПК на основе ранее созданных компонентов. Вместе с тем эта модель должна получить широкое распространение, поскольку сборка систем из готовых или ранее использованных компонентов значительно ускоряет разработку ПО.

Итерационные модели процесса создания ПК являются альтернативой описанному и позволяют при изменении требований повторно выполнять ряд этапов по созданию системы.

Наиболее используемыми итерационными моделями являются модель пошаговой разработки и спиральная модель. В модели пошаговой разработки процессы специфицирования требований, проектирования и написания кода разбиваются на последовательность небольших шагов, которые ведут к созданию ПК. Спецификация ПК определяет все функции и действия, которые будет выполнять разрабатываемая система. В спиральной модели весь процесс создания ПК разворачивается по спирали, от начального эскиза системы до ее конечной реализации.

Существенным отличием итерационных моделей является то, что здесь процесс работы спецификации протекает параллельно с разработкой самой программной системы. Более того, в модели пошаговой разработки полную системную спецификацию можно получить только после завершения самого последнего шага процесса создания ПК. Очевидно, что такой подход входит в противоречие с моделью приобретения ПО, когда полная системная спецификация является частью контракта на разработку системы, и требует изменения формы контракта, на что крупные организации идут с трудом.

Выделение нерешенных ранее частей общей проблемы

Рассмотренные модели процесса разработки ПК имеют ряд преимуществ и недостатков. При создании больших систем используются различные подходы к разработке разных частей системы, и в целом к разработке системы применяются смешанные модели [2]. Очевидно, что существующие на текущий момент модели не могут быть применены «в чистом виде» к процессу создания любого программного комплекса. Исходя из этого, возникает необходимость в разработке новой обобщенной модели создания ПК.

Формулирование целей статьи

Таким образом, в рамках данной работы поставлена задача на основании анализа недостатков рассмотренных моделей процесса создания ПК предложить новую обобщенную модель, а также разработать алгоритм создания программного комплекса на основании предложенной модели.

Изложение основного материала

Схема предложенной обобщенной модели процесса создания программного комплекса представлена на рис. 1. Эта модель представляет собой наиболее полное представление процесса проектирования ПК с практической точки зрения и учитывает наиболее важные практические аспекты данного процесса.

Процесс разработки ПК АС можно условно разделить на этапы. Подобная декомпозиция дает возможность некоторым образом сгруппировать операции, выполняемые при разработке ПК АС, выделить и описать последовательность их выполнения, тем самым определив эту последовательность операций как один из возможных алгоритмов выполнения данного этапа проектирования АС.

На этапе «*Определение требований к ПО*» осуществляется поиск, формализация и формулировка требований, которые заказчик предъявляет к программному обеспечению.

Способ ведения разработки функционального ПО (ФПО) итеративным способом, с возвратом к этапу определения требований, используется многими технологиями и схемами создания ФПО, например, технологией, получившей название «*Экстремальное программирование*» (XP).

Этап «*Общее проектирование структуры ПО*» предназначен для детализации проработки общей структуры ФПО, а именно проработки количества и состава модулей, описания их взаимодействия, а также возможностей стыковки с модулями, разработка и внедрение которых возможны лишь в перспективе.

На основании анализа и работ, выполненных на данном этапе, осуществляется выбор варианта структуры программной системы. При разработке корпоративных программных систем, а также систем, включающих большое количество пользователей, применяется распределенная структура. Для систем

меньшого масштаба или для корпоративных систем, требующих жесткой централизации, применяется клиент-серверная структура. Для небольших программных систем, решающих прикладные задачи и не требующих организации взаимодействия с другими задачами, применяется локальная структура программной системы.

На этапе «Анализ структурных компонентов ФПО» осуществляется «распределение» требований, предъявляемых к прикладному ПО в целом, применительно к компонентам (модулям) структуры ФПО, полученной и утвержденной на предыдущем этапе. На основании требований непосредственно к отдельным компонентам ФПО может быть осуществлена детализация этих компонентов в рамках уже разработанной структуры.

Результаты данного этапа оформляются в виде детализированной схемы, которая получена на предыдущем этапе, с применением CASE-средств.

На этапе «Анализ возможности применения существующих программных решений в рамках разрабатываемой системы» осуществляется поиск и анализ возможности применения в составе ФПО разрабатываемой системы существующих программных реализаций каких-либо аналогичных решений.

На этапе «Выбор системного ПО под различные компоненты» осуществляется анализ структуры ФПО, разработанной на предыдущих этапах, покомпонентно. Для каждого компонента, обладающего самостоятельностью в рамках локальной компьютерной сети, необходимо осуществить выбор операционной системы и другого системного ПО.

На этапе «Разработка компонентов ФПО создаваемой системы» осуществляется непосредственная разработка программной реализации компонентов системы. При этом разработка каждого компонента осуществляется индивидуально, однако в реализации компонента уже на данном этапе закладываются механизмы взаимодействия с другими компонентами.

Параллельно с ведением непосредственной разработки программного продукта может также осуществляться разработка тестов для него, а также применяться рефакторинг. Под рефакторингом подразумевается переработка кода для повышения эффективности общего процесса разработки, а также для снижения затрат на этапе сопровождения. Разработка тестов позволяет создавать базу тестов, которая будет применяться на последующих этапах, параллельно с формированием бизнес-логики программной реализации. Ведение параллельного рефакторинга позволяет команде разработчиков осуществлять более эффективную работу, а также значительно снизить трудозатраты на последующих этапах.

Тем не менее, параллельная разработка тестов и ведение рефакторинга не являются обязательными и могут не выполняться при недостатке времени.

На этапе «Синхронизация компонентов ФПО» осуществляют программную доработку компонентов ФПО, но уже не в рамках их непосредственной функциональной нагрузки, а для организации эффективной совместной работы с другими компонентами. То есть на данном этапе осуществляется разработка и синхронизация механизмов взаимодействия различных компонентов, заложенных на предыдущем этапе, разработка и описание набора правил взаимодействия этих компонентов, совокупность которых представляет собой протокол межкомпонентного взаимодействия.

Работы по синхронизации компонентов программной системы осуществляются в том случае, когда применяется распределенная структура.

При применении клиент-серверной структуры программной системы основная задача состоит лишь в разработке протокола клиент-серверного взаимодействия, реализация которого закладывается как в клиентскую, так и в серверную часть программного комплекса, и дальнейшая их синхронизация не требуется.

На этапе «Тестирование компонентов ФПО» осуществляется тестирование каждого отдельного компонента на предмет наличия ошибок в его функциональности. Тестирование компонентов дает возможность на данном этапе выявить основные ошибки, допущенные при реализации функциональности компонентов, пока не появились ошибки, связанные с системной характеристикой всего комплекса ФПО. То есть наиболее желателен вариант сбора комплекса ФПО из абсолютно (в идеале) безошибочных компонентов. В этом случае, с одной стороны, уменьшится общий объем ошибок, обнаруживаемых при целостном тестировании системы, а с другой стороны, ошибки, допущенные в функциональности компонентов, не будут порождать новые общесистемные ошибки.

Тестирование комплекса ФПО в целом осуществляется на этапе «Тестирование ПО АС». При этом предполагается, что недочеты, которые возможно было устранить непосредственно на уровне компонентов, были исправлены на предыдущем этапе, а упор делается на тестирование системы в целом.

На этапе *рефакторинга* осуществляется приведение уже готового кода к более удобочитаемому виду. При этом осуществляется не столько форматирование кода, сколько его переработка. Кроме того, проведение этой операции дает возможность некоторым образом нормализовать код.

Под нормализацией в данном случае понимается придание коду некоторой «структуры», которая, с одной стороны, интуитивно понятна другим разработчикам, а с другой стороны, введение этой структуры исключает ряд негативных моментов, таких как, например, дублирование кода.

На этапе «Анализ производительности комплекса ФПО АС» осуществляется оценка времени выполнения различных операций компонентами системы и сопоставление их с исходными требованиями.

После анализа производительности в рамках этапа «Оптимизация быстродействия комплекса ФПО АС» осуществляется доработка функций, выявленных на предыдущем этапе. То есть осуществляется изменение ис-

ходного кода, поиск и реализация более быстродействующих решений и алгоритмов, устранение ошибок, замедляющих работу комплекса ФПО.

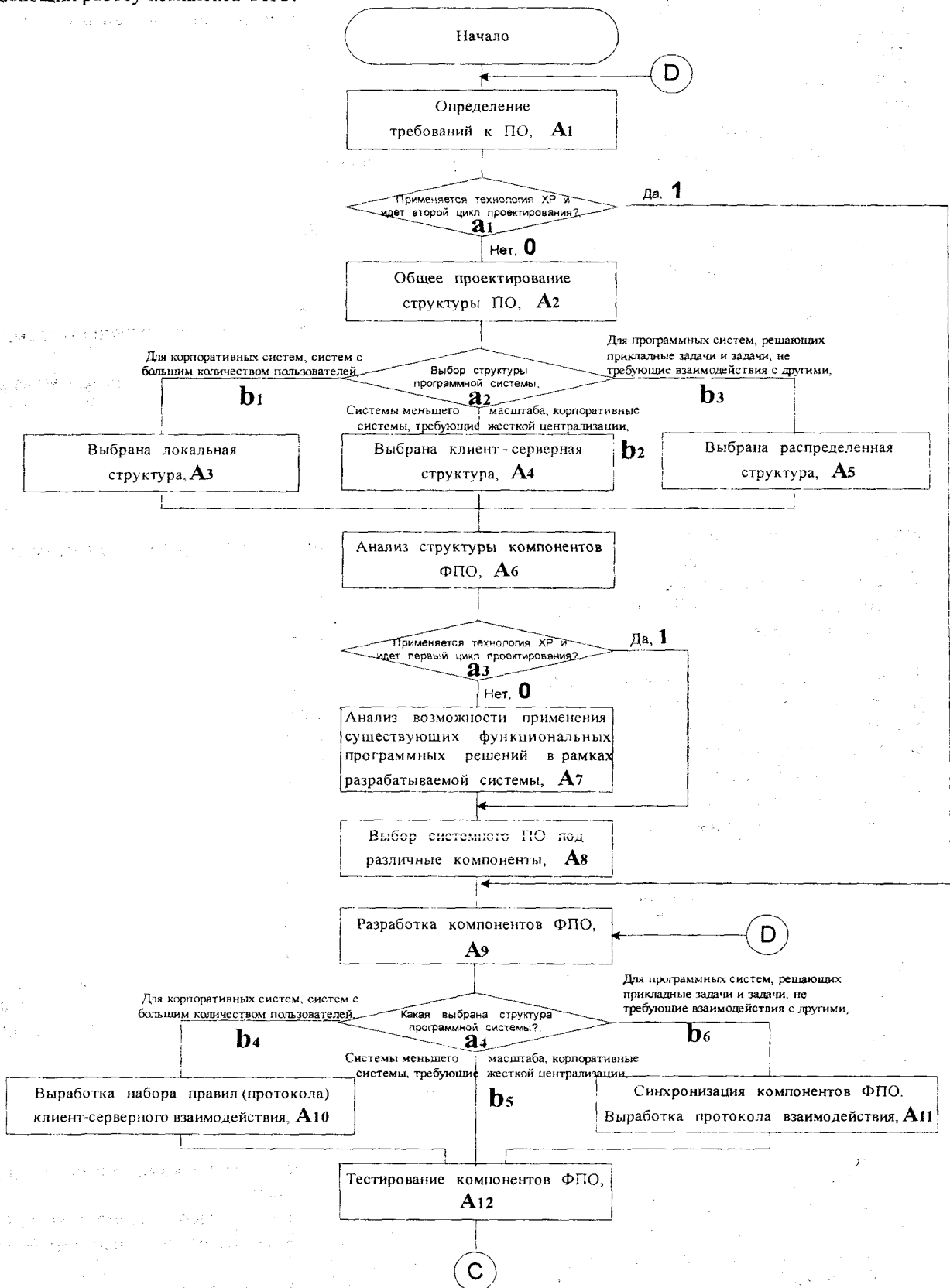


Рис. 2. Схема обобщенного алгоритма создания программного комплекса автоматизированных систем.

На етапі «*Разработка технической документации программной системы*» іде підготовка і формування різних документів, супроводжуваних програмний продукт. Розробка технічної документації дозволяє не тільки доповнити програмне рішення необхідними документами і превратити його в продаваний продукт, але і зафіксувати рішення (схеми), які були створені в ході розробки ПО.

Набір технічної документації розробника представляє собою набір формалізованих рішень, прийнятих в ході розробки, і служить джерелом інформації для доработки і розвитку системи. Ця документація використовується при ітеративній схемі ведення розробки, а також на етапі супроводження системи.

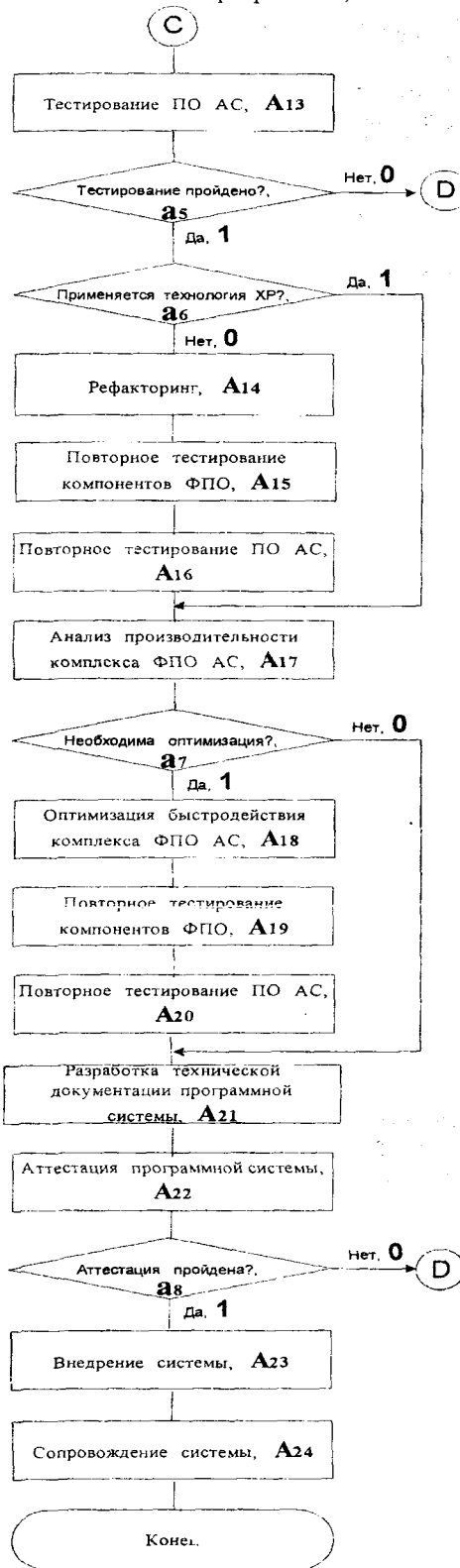


Рис. 2. (продолжение). Схема обобщенного алгоритма создания программного комплекса автоматизированных систем.

На етапі «Аттестация программной системы» проводятся мероприятия по проверке соответствия созданного программного продукта требованиям заказчика. При использовании различных схем разработки ПО многие из них включают проведение тестирования на различных уровнях декомпозиции программной системы. На основании предложенной схемы основной задачей данного этапа является только проверка соответствия программной реализации выдвинутым требованиям, так как тестирование системы было осуществлено на предыдущих этапах.

При внедрении разрабатываемой системы осуществляется организация мероприятий по подготовке оборудования, персонала и объекта автоматизации к вводу системы в эксплуатацию.

В рамках сопровождения создаваемой системы осуществляются действия по ее поддержке в работоспособном состоянии, дополнительной настройке, а также по предотвращению морального износа системы.

На основании предложенной обобщенной схемы процесса разработки ПК был разработан алгоритм, позволяющий четко регламентировать последовательность действий разработчика при выполнении работ по созданию ПК. Схема данного алгоритма приведена на рис. 2. Этот алгоритм представляет собой детальное описание последовательности этапов создания программного комплекса АС. Данный алгоритм позволяет осуществить выбор и анализ методологии проектирования, детально описать процесс создания ПК, а, следовательно, дать ориентировочные оценки длительности, трудоемкости и других параметров данного процесса.

Этот алгоритм был представлен с использованием регулярного выражения алгоритма (РВА), то есть уравнения алгоритма, составленного на основе алгебры событий [3, 4].

Модель создания программного комплекса автоматизированных систем представлена в следующем виде:

$$\begin{aligned} & \{ \{ A_1 \cdot ([a_1 = 0]) \cdot (A_2 \cdot ([a_2 = b_1]) \cdot A_3 \vee [a_2 = b_2]) \cdot A_4 \vee [a_2 = b_3]) \cdot A_5 \cdot ([a_3 = 0] \cdot A_7 \cdot A_8) \} \wedge \\ & \wedge \{ \{ A_9 \cdot ([a_4 = b_4]) \cdot A_{10} \vee [a_4 = b_5]) \cdot A_{11} \cdot A_{12} \cdot A_{13} \} \cdot [a_5 = 0] \} \cdot \{ \{ a_6 = 0 \} (A_{14} \cdot A_{15} \cdot A_{16}) \} \wedge \\ & \wedge A_{17} \cdot ([a_7 = 1]) \cdot (A_{18} \cdot A_{19} \cdot A_{20}) \cdot A_{21} \cdot A_{22} \} \cdot [a_8 = 0] \} \cdot A_{23} \cdot A_{24}, \end{aligned}$$

где

A_i – этапы или задачи в рамках создания ПК системы,

a_j – условия, определяющие направления проектирования,

b_k – варианты решений, соответствующие различным условиям.

Выводы из данного исследования

Таким образом, на основании анализа существующих моделей процесса разработки ПК было выполнено описание процесса создания программного комплекса АС и разработка его обобщенной модели, учитывающей наиболее актуальные и применяемые на практике этапы, а также наиболее применяемые способы ведения разработки.

На основании данной модели был создан обобщенный алгоритм ведения разработки ПК, а также разработано его формализованное представление с использованием регулярных выражений алгоритмов. Перевод предложенной модели создания ПК АС в аналитическую область посредством РВА позволил в большей степени формализовать данную модель и обеспечить возможность применения к ней математических методов.

ЛИТЕРАТУРА:

1. Фатреля Роберт Т., Шафер Дональд Ф., Шафер Линда И. Управление программными проектами: достижение оптимального качества при минимуме затрат. – М.: Вильямс, 2003. – 1136 с.
2. Буч Гради, Рамбо Дж., Якобсон А. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.
3. Глушков В.М. Синтез цифровых автоматов. – М.: Физматгиз, 1962. – 476 с.
4. Глушков В.М. Кибернетика, вычислительная техника, информатика // Избранные труды. Т. 1: Математические вопросы кибернетики. Некоторые проблемы синтеза цифровых автоматов. – К.: Наукова думка, 1990. – 262 с.

Получено редакцией 25.02.2005.

© Левыкин В.М., 2005.

© Жигалов М.А., 2005.

Левыкин Виктор Макарович, доктор технических наук, профессор.

Жигалов Максим Александрович, студент.

Харьковский национальный университет радиоэлектроники.

станку САШ-420М / М.О. Ткаченко // Нові технології. – 2005. - № 1-2 (7-8). – С. 76.

Розглянуто проблеми визначення кінематичних параметрів шліфування кремнієвих пластин на станку САШ-420М. Запропоновано процедуру оптимізації параметрів шліфування пластин. Розроблено алгоритм управління параметрами процесу шліфовки кремнієвих структур з діелектричною ізоляцією.

Лл. 8. Бібліогр.: 5 назв.

УДК 621.382

Виготовлення епітаксійних структур для малошумових біполярних транзисторів / С.О. Келеберденко, Л.О. Серов // Нові технології. – 2005. - № 1-2 (7-8). – С. 86.

Проведено порівняльний аналіз електрофізичних і шумових характеристик біполярних транзисторів, при виробництві яких епітаксійний шар формувався на різних установках епітаксійного нарощування. Транзистори з епітаксійним шаром, осадженим на установці «Епіквар 101м», мають значення коефіцієнта шуму нижче середнього, і вище - коефіцієнт виходу придатних до використання приладів по ВАХ, у порівнянні з установками «УНЕС 101» і «УНЕС 2ПКА».

Табл. 1. Бібліогр.: 3 найм.

УДК 658.012.011.56

Методи та засоби розробки логічних структур баз даних web-орієнтованих інформаційних систем / В.М. Левикін, О.В. Петриченко, А.О. Петренко // Нові технології. – 2005. - № 1-2 (7-8). – С. 88.

У статті пропонується модифікація існуючої методики проектування логічних структур баз даних, а також реалізовані елементи інструментального засобу – системи підтримки прийняття рішень, супроводжуючих процес проектування баз даних web-орієнтованих інформаційних систем. Запропонована методика дозволяє суттєво скоротити терміни розробки структур баз даних таких систем, забезпечує отримання раціональних проектних рішень.

Лл. 2. Бібліогр.: 3 найм.

УДК 62-55:681.515

Оцінка робастності системи управління температурою газу двовального двоконтурного двигуна // В.І. Гостєв, А.М. Чаузов, С.С. Баранов // Нові технології. – 2005. - № 1-2 (7-8). – С. 92.

Викладена оцінка робастності системи управління температурою газу двовального двоконтурного газотурбінного двигуна. Представлені результати математичного моделювання системи з цифровим оптимальним за швидкодією регулятором та нечітким (що працює на базі нечіткої логіки) регулятором.

Лл. 6. Бібліогр.: 3 найм.

УДК 621.313.33

Розробка САПР для дослідження характеристик електротехнічних пристроїв з метою їх оптимізації на основі числових методів / О.В. Качура, С.В. Количев, О.М. С'янов // Нові технології. – 2005. - № 1-2 (7-8). – С. 98.

В статті наведена модель асинхронного двигуна з масивним феромагнітним ротором, розглянута методика розрахунку електромагнітного поля методом скінчених елементів. Розроблена інтерактивна програмна система, яка дозволяє досліджувати поле різноманітних електротехнічних пристроїв.

Лл. 12. Бібліогр.: 5 найм.

УДК 658.012.011.56

Розробка узагальненої моделі створення програмного комплексу автоматизованих систем / В.М. Левикін, М.А. Жигалов // Нові технології. – 2005. - № 1-2 (7-8). – С. 106.

Вирішена і обґрунтована задача створення узагальненої моделі процесу розробки програмного комплексу АС. В рамках цієї задачі проведено аналіз існуючих моделей розробки програмного комплексу, за результатами якого виявлені їх переваги та недоліки, та запропонована нова узагальнена модель. Запропоновано алгоритм створення програмного комплексу

machine SDP-420M / Tkachenko M.O. // New technology. – 2005. - № 1-2 (7-8). – P. 76.

The paper considered problems determination of kinematics parameters silicon wafer polishing on a machine SDP-420M. A procedure optimization of parameters wafers polishing is offered. An algorithm control by parameters process of polishing silicon structures with the dielectric isolation is developed.

Fig. 8. Ref.: 5 items.

UDC: 621.382

Manufacturing of epitaxial structures for low-noise bipolar transistors / E.O. Keleberdenko, L.O. Serov // New technology. – 2005. - № 1-2 (7-8). – P. 86.

Comparative analysis of bipolar transistors electrophysical and noise characteristics is carried out; during analysis epitaxial layer was formed on different epitaxial growth units. Transistors with epitaxial layer, precipitated on «Epiquar 101m» unit, have lower noise average value coefficient and higher effective devices outlet coefficient according to E-I characteristics, in comparison with «UNES 101» and «UNES 2PKA» units.

Tabl. 1. Ref.: 3 items.

UDC: 658.012.011.56

Methods and tools of database logical structures development of web-oriented informational systems / Levikin V.M., Petrichenko A.V., Petrenko A.A. // New technology. – 2005. - № 1-2 (7-8). – P. 88.

In this work offered the modification of existing database logical structures development methodic. Also developed and implemented the elements of the expert support system that maintains the process of web-oriented informational systems' databases development. The proposed methodic allows to reduce time spent to database development of the web-oriented informational systems, providing the rational project solutions.

Fig. 2. Ref.: 3 items.

UDC: 62-55:681.515

Robustness estimates of gas temperature control system of two-spool and double-loop gas-turbine engine / V. Gostev, A. Chausov, S. Baranov // New technology. – 2005. - № 1-2 (7-8). – P. 92.

Robustness estimates outcomes of gas temperature control system of two-spool and double-loop gas-turbine engine. Representation of mathematic modulate system with digit optimal speed regulator and indistinct (working on indistinct logic base) regulator.

Fig. 6 Ref.: 3 items.

UDC: 621.313.33

Development of the application package for account of an electromagnetic field / A.V. Kachura, S.V. Kolichev, A.M. Syanov // New technology. – 2005. - № 1-2 (7-8). – P. 98.

In the article the model of an asynchronous drive with a massive ferromagnetic curl is indicated, the technique of account of an electromagnetic field by a finite element method is considered. The interactive program system, permitting to investigate a field of various electrotechnical devices is developed.

Fig. 12. Ref.: 5 items.

UDC: 658.012.011.56

Development of generalized model of computer-aided systems' bundled software creation / Levikin V. M., Zhigalov M. A. // New technology. – 2005. - № 1-2 (7-8). – P. 106.

The task of creation of generalized model of computer-aided systems' bundled software was solved and based. In the context of this task the analysis of existent models for bundled software creation was made. Basing on results of the analysis advantages and disadvantages were found out, and the new generalized model was suggested. The algorithm of creation of computer-aided systems'