

для параметра i и 0 в противном случае. После вычислений мы получим результирующую матрицу $R = \{r_k^i\}$, где r_k^i - результат вычислений для свойства i с применением метода k : $r_k^i = F(V^k, mp_k^i, m_i)$.

Выводы. В результате проведенных исследований показаны методологические основы необходимости и целесообразности применения ИТЭП в общей схеме МТАП сложных ИУС, создаваемых с использованием КНР. Представлены также разработанные модели, позволяющие формализовать все основные этапы ИТЭП. Научная новизна данного подхода заключается в применении современных методов прототипирования к области исследования атрибутов качества ПО. Практическая значимость результатов состоит в повышении эффективности процессов создания ИУС путем сокращения затрат ресурсов и повышения качества получаемых проектных решений. Перспективной дальнейших исследований в этом направлении является совершенствование полученных моделей и разработка методики многокритериального сравнения эффективности различных способов обеспечения атрибутов качества КНР.

Список литературы: 1. Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки. – СПб.: Питер, 2005. – 412 с. 2. Архитектуры, модели и технологии программного обеспечения информационно-управляющих систем / Н.В. Ткачук, В.А. Шеховцов, Д.В. Кукленко, В.Е. Сокол / Под ред. М.Д. Годлевского. – Харьков: НТУ «ХПИ», 2005. – 546 с. 3. Ткачук Н.В., Годлевский М.Д. Адаптивная технология разработки и реинжиниринга сложных ИУС в среде многомерного информационного метaprостранства // Труды 7-й международной научно-практической конференции «Современные информационные и электронные технологии», 22-26 мая 2006 г., г. Одесса. - Т.1 - С. 13. 4. Ткачук Н.В., Земляной А.А., Гамзаев Р.А. Технология эволюционного прототипирования компонентных программных решений для информационно-управляющих систем. // Тезисы докладов IX международной научно-технической конференции «Системный анализ и информационные технологии», НТУУ «КПИ», Киев, 15-19 мая 2007 г. - с. 206. 5. Luqi, R., Steigerwald, G., Hughes, V., Berzin A. CAPS as a Requirements Engineering Tool.. Proc. Conference on TRI-Ada, San Jose, CA, 75-83, 1991. 6. Herranz, A, Moreno-Navarro, J. Rapid Prototyping and Incremental Evolution Using SLAM, Proc. 14th IEEE International Workshop on Rapid Systems Prototyping, San Diego, 201-209, 2003. 7. Axure Software Solutions, <http://www.axure.com/products.aspx>. 8. iRise Company, <http://www.irise.com/>. 9. Ткачук Н.В., Гамзаев Р.А. Нечеткий подход к решению задачи анализа системных требований в процессах разработки и сопровождения информационных систем // Автоматизированные системы управления и приборы автоматизации. – Харьков: ХНУРЭ. – 2006. - Вып. 134. - С. 64- 70. 10. Ткачук Н.В., Горелый А.В., Земляной А.А. Комплекс имитационных моделей для исследования компонентных программных решений в ИУС АСУ ТП // Вісник Національного технічного університету "ХПИ". - Харків: НТУ "ХПИ". – 2004. - № 18, – С.145-152.

Поступила в редакцию 24.02.08

УДК 004.4'22

В.В. ЛАНИН, Пермский государственный университет,
г. Пермь, Россия
Л.Н. ЛЯДОВА, канд. физ.-мат. наук, АНО науки и образования
“Институт компьютеринга”, г. Пермь, Россия

ИНТЕЛЛЕКТУАЛЬНЫЕ CASE-СРЕДСТВА ДЛЯ РАЗРАБОТКИ АДАПТИРУЕМЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Представлено підхід до створення інтелектуальних інформаційних систем, що є основою CASE-технології METAS, призначеної для підтримки життєвого циклу динамічно адаптуємих розподілених інформаційних систем. Системи, створені за допомогою CASE-засобів METAS, функціонують у режимі інтерпретації. Вони базуються на багаторівневих мета-даних, які описують предметну область системи. Можливості адаптації реалізуються через засоби реструктуризації даних, налаштування користувальницького інтерфейсу, генерації запитів і звітів, динамічного підключення програмних компонентів сторонніх розроблювачів. Одна з найбільш важливих - підсистема керування документами, що забезпечує можливості інтелектуального пошуку, аналізу автоматичної класифікації й каталогізації документів, заснована на використанні онтологій й агентного підходу.

Представлен подход к созданию интеллектуальных информационных систем, являющийся основой CASE-технологии METAS, предназначенной для поддержания жизненного цикла динамически адаптируемых распределенных информационных систем. Системы, созданные с помощью CASE-средств METAS, функционируют в режиме интерпретации. Они базируются на многоуровневых метаданных, которые описывают предметную область системы. Возможности адаптации реализуются через средства реструктуризации данных, настройки пользовательского интерфейса, генерации запросов и отчетов, динамического подключения программных компонентов сторонних разработчиков. Одна из наиболее важных – подсистема управления документами, обеспечивающая возможности интеллектуального поиска, анализа автоматической классификации и каталогизации документов, основанная на использовании онтологий и агентного подхода.

An approach to development of intelligent information systems is presented. This approach is the base of CASE-technology METAS intended for support of adaptable distributed information systems life cycle. Information systems created with METAS CASE-tools operates in interpretation mode. It's based on multilayer metadata which describes universe of discourse for the information system. The adaptation capabilities are realized by means of data restructuring tools, interface generation and tuning, query builder and reports generation, dynamic linking of program components developed by exterior application designer. Document management subsystem is one of the most important. It includes tools of storing documents in database, business intelligent means, automatic classification and intelligent search of electronic documents. This subsystem is based on ontologies and multi-agents architecture.

Введение. Адаптируемость является важнейшим свойством современных информационных систем (ИС), которое проявляется как нефункциональ-

ные требования для различных классов информационных систем, систем e-learning и электронной коммерции и т.п.

Все программные системы функционируют в определенном окружении, поэтому адаптируемые системы должны настраиваться на изменения среды, чтобы обеспечить развитие и повысить «живучесть». Понятие адаптируемости рассматривается очень широко, в него включаются такие взаимосвязанные требования, как гибкость, расширяемость, способность к развитию, интероперабельность систем и переиспользуемость компонентов и т.п. Это делает адаптируемость систем не только интересным, но и критичным свойством в практике разработки новых систем и сопровождения существующих. Адаптируемость обеспечивает более простое и эффективное сопровождение и разработку программного обеспечения. Настройка программных систем является неизбежным процессом, обеспечивающим соответствие меняющимся потребностям пользователей, условиям эксплуатации.

Таким образом, существует необходимость в создании технологии, обеспечивающей соответствие требованиям адаптируемости. Первым шагом в создании программного решения является разработка архитектуры системы, которая позволяла бы адаптировать к меняющимся условиям как используемые для разработки инструментальные средства, так и создавать с их помощью адаптируемые информационные системы. Далее технология создания информационных систем должна поддерживать изменения модели предметной области ИС в соответствии с новыми требованиями, появляющимися в процессе ее эксплуатации. Одной из важнейших сторон создания адаптируемых систем является обеспечение пользователей средствами управления интерфейсом, его настройкой.

В данной статье описывается технология, которая обеспечивает разработчиков и пользователей средствами для эффективной настройки информационных систем, созданных на базе этой технологии.

1. Технология создания адаптивных систем. Средства, используемые для создания крупных распределенных информационных систем, состоящих из отдельных территориально распределенных подсистем, должны удовлетворять требованиям, обеспечивающим возможности их настройки на различные условия эксплуатации и потребности пользователей как в процессе установки, так и динамически, в ходе функционирования ИС.

Если информационная система создается для установки в организациях, являющихся бизнес-партнерами, но обладающих различными техническими возможностями, занимающихся различными видами деятельности и т.п., то процесс ее разработки и сопровождения значительно ус-

ложняется. Реализация перечисленных выше требований обеспечивает эффективность вложений в создание и сопровождение системы, повышает ее живучесть.

Максимальная адаптируемость ИС может быть достигнута, если как при разработке системы, так и в ходе ее эксплуатации применяются *метамодели* и *метаданные*, описывающие особенности предметной области, для которой создается система, условия ее работы.

CASE-технология METAS (METAdata System), разработанная АНО «Институт компьютеринга», – это основа для создания систем, управляемых метаданными. Ключевым моментом технологии является использование *взаимосвязанных многоуровневых метаданных*, описывающих информационную систему, создаваемую для определенной предметной области, с различных точек зрения и на разных уровнях детализации.

Основное отличие данной CASE-технологии от многих существующих, генерирующих код приложений на каком-либо языке программирования по некоторым спецификациям (метаданным), описывающим предметную область, состоит в том, что данная система использует это описание *во время своей работы*, выполняя функции отображения данных, их поиска и обработки по определенным этими метаданными правилам, *интерпретируя* их [1].

Применение метаданных в режиме интерпретации дает возможность *гибкой динамической настройки приложения*, его функциональности, реструктуризации информационных объектов, описанных метаданными.

Кроме того, при таком подходе к разработке ИС проект обладает высокой степенью «обратной связи», так как разработчик, меняя метаданные, сразу видит соответствующие изменения в ИС, реализуемой на основе данной технологии (в ее информационных объектах и связях между ними, интерфейсе пользователя и т.п.). Разработчик фактически работает с той же системой, что и пользователь, но он имеет права на использование специального CASE-инструментария системы, как при создании ИС, так и в ходе ее эксплуатации.

Это создает хорошие предпосылки для создания «интеллектуальной» системы, которая может настраиваться на потребности пользователя и меняющиеся условия эксплуатации *в ходе работы с ней пользователей*.

Метаданные – это *формализованное описание ИС*, используемое для настройки приложения на условия его эксплуатации в процессе его разработки, а затем – загрузки и выполнения. Метаданные описывают объекты предметной области ИС и их поведение; визуальный интерфейс пользователя ИС; бизнес-операции и бизнес-процессы предметной области; первичные и сводные документы и отчеты и т.п. Метаданные представляют

модели (рис. 1), каждая из которых описывает определенную часть, «срез» ИС (некоторые модели могут описывать одни и те же части ИС, но с различных точек зрения).

Модели разделены на *слои*, образующие иерархию. Их описание хранится в базе метаданных (БМД) системы. Основные модели METAS:

- *Физическая модель* (Physical Model) – представлена метаданными, описывающими представление объектов ИС (описание таблиц, в которых хранятся данные об объектах, связей между ними и пр.) в базе данных (БД).
- *Логическая модель* (Logical Model) – описание сущностей предметной области, для которой создается ИС, их поведения (через операции), а также общие операции ИС. Данная модель основывается на нотациях языка UML.
- *Презентационная модель* (Presentation Model) описывает визуальный интерфейс пользователя при работе с объектами ИС.
- *Модель репортинга* (Reporting Model) – представлена с помощью метаданных, описывающих запросы, шаблоны документов и отчеты, формируемые в ходе выполнения бизнес-операций и бизнес-процессов, используемые для визуализации и анализа данных ИС.
- *Модель бизнес-процессов* (Workflow Model) – описание бизнес-операций и бизнес-процессов, поддерживаемых ИС. Бизнес-процессы включают бизнес-операции и стандартные блоки управления, объединяющие их. С бизнес-процессами связываются потоки данных, документов, которые формируются и обрабатываются при выполнении бизнес-процессов.

Модели могут изменяться и расширяться при создании новых компонентов ИС (например, Web-компонентов), или изменении существующих.

Web-модель обеспечивает доступ к ресурсам ИС для удаленных пользователей через Web-интерфейс.

Отдельная модель используется для обеспечения функционирования *средств защиты* ИС. *Модель защиты* (Security Model) представляет пользователей системы и их права на доступ к БД, выполнение операций над объектами ИС или на доступ к моделям метаданных.

Программные компоненты ИС работают с моделями соответствующего уровня (или нескольких взаимосвязанных уровней). Метаданные позволяют представить свойства, взаимосвязь и взаимодействие компонентов системы.

Компоненты интеграции – это NET-компоненты, которые подключаются к системе и получают доступ к моделям (метаданным). Они могут подписываться на различные события и расширять функциональность

через добавление нестандартных действий. Кроме общих компонентов интеграции используются также пользовательские *бизнес-операции*, *триггеры* и *ограничения*, позволяющие добавлять в приложения нестандартную, специфичную для конкретной ИС логику.

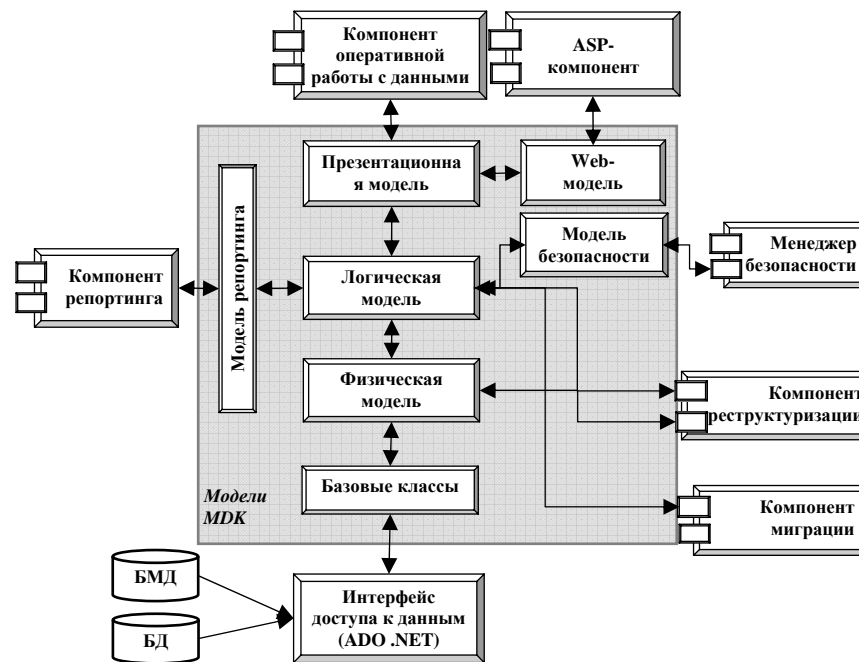


Рис. 1. Модели метаданных и программные компоненты METAS

Система реализована на платформе DOT NET Framework. Для работы с СУБД используется технология ADO.NET. Клиент-серверное взаимодействие основано на технологии DOT NET Remoting.

Использование CASE-технологии METAS позволяет создать ИС, архитектура которой представляет собой клиент-серверное приложение, разбитое на *домены*. Разбивка на домены ориентирована на реализацию распределенных ИС, включающих автономно функционирующие, не имеющие оперативной связи подсистемы с архитектурой клиент-сервер. Интеграция доменов осуществляется на основе *технологии BizTalk* (сервер BizTalk является отдельным приложением). Для связи BizTalk-серверов может использоваться любой транспортный протокол, поддерживаемый

конкретным сервером BizTalk (обычно список протоколов достаточно широк).

2. Система управления документами и динамическая настройка информационных систем. Подсистема управления документами CASE-системы METAS включает средства репортинга [2], поиска, анализа и классификации и каталогизации документов [3].

Одной из проблем, препятствующих полномасштабному переходу к электронным документам, является отсутствие в современных системах управления документами средств, позволяющих работать с электронными документами с учетом семантики их содержания. На сегодняшний день фактически все преимущества электронных документов вытекают только из формы их представления (удобство хранения, поиска по ключевым словам, копирования, передачи и т.п.). С других точек зрения (возможность автоматизации обработки, классификации, каталогизации, анализа и пр.) большинство электронных документов практически аналогичны по возможности их использования традиционным «бумажным» документам, то есть содержание документа может быть воспринято и обработано только человеком («документ для человека»).

Большинство документов ориентировано на восприятие человеком, так как в них отсутствует дополнительная информация об их структуре и содержании, которая может помочь в автоматическом извлечении информации компьютером. Из сказанного выше следует, что извлечение информации из слабоструктурированных документов на сегодняшний день является чрезвычайно актуальной и, в то же время, сложной задачей. Таким образом, необходимы дополнительные средства описания семантики информации, содержащейся в документах.

Современный электронный документ кроме собственно данных требует наличия метаданных, описывающих структуру документов и семантику содержащихся в них данных. За счет такого подхода эффективность работы с электронными документами может быть резко повышена, т.к. станет возможным *автоматический анализ информации, содержащейся в документе* («документ для человека и для системы»).

Одним из возможных решений указанных выше проблем может служить комплексный подход, основанный на методах искусственного интеллекта и различных парадигмах представления знаний [3, 4].

Актуальность разработки интеллектуальных систем, решающих задачи управления документами на основе комплексного подхода, интегрированных в информационные системы, автоматизирующие деятельность сотрудников предприятий, организаций (бизнес-систем) при выполнении

ими типовых операций (бизнес-операций) и аналитической обработке данных, вызвана следующими факторами:

- Получение *результатов обработки данных*, хранящихся в базе данных (БД), хранилище данных – одна их основных информационных потребностей пользователя любой информационной системы. Следовательно, нужны средства *формирования запросов* для получения необходимой пользователю информации из базы данных или хранилища. По возможности *запросы должны формироваться в терминах предметной области*, в которой работает пользователь.
- Результаты, получаемые на основе данных, должны быть определенным образом *представлены* – выведены в форме документов различных форматов в зависимости от потребностей пользователей. Следовательно, нужны *средства подготовки и генерации отчетов*, по возможности, не требующие навыков программирования, доступные пользователям, умеющим работать в среде современных офисных продуктов.
- Пользователи ИС имеют дело с множеством документов, как создаваемых в самой системе описанными выше средствами, так и получаемых в различных форматах из внешних источников. Отсюда следует необходимость решения задачи их *унифицированного хранения, классификации и каталогизации в базе данных ИС*, позволяющей выполнять эффективный поиск документов пользователями, а также их последующую обработку, анализ.
- Деятельность любой бизнес-системы строится на основе нормативных документов. Поддержка бизнес-операций средствами информационной системы требует *отражения в модели данных информационной системы норм, закрепленных в нормативно-справочных данных, распорядительных документах*, в виде ограничений, налагаемых на данные (атрибуты, свойства объектов предметной области, информация о которых хранится в БД, а также связи между ними) и операции, выполняемые над ними. Эти ограничения вносятся в модель данных ее разработчиками, которые должны при создании модели выполнить анализ предметной области (а это, прежде всего, – *анализ документов*, регламентирующих деятельность бизнес-системы) и построить в результате этого анализа модель предметной области или обновить ее в соответствии с новыми требованиями, условиями деятельности системы, которые меняются в результате изменений нормативной базы. Если поставить задачу динамической настройки информационной системы на меняющиеся условия, то основа реализации средств ее динамической адаптации – средства

реструктуризации данных в БД ИС. А эти средства позволяют вносить изменения в модель данных на основе результатов *анализа предметной области, нормативно-справочных и распорядительных документов, регламентирующих деятельность в этой области*. Отсюда следует необходимость поддержки в динамически адаптируемых системах одного из самых сложных и трудоемких этапов разработки ИС – этапа анализа.

Поддержка возможности динамической адаптации может быть основана на средствах анализа документов (в частности, нормативно-справочных) [4]. В результате анализа должна быть построена система *взаимосвязанных документов*:

- относящихся к определенным направлениям деятельности бизнес-системы (к определенным понятиям, объектам предметной области);
- отражающих связи между этими понятиями (с каждым понятием может быть связан документ или совокупность документов, связи между документами отражают связи между понятиями);
- содержащих нормативную информацию, которая также может быть выделена на основе анализа содержания документов.

На основе построенной системы взаимосвязанных документов можно *частично автоматизировать процесс анализа изменений предметной области и внесения изменений в модель предметной области ИС* (т.е. реализовать поддержку процесса разработки и адаптации ИС).

Таким образом, система управления документами становится не только «надстройкой» над ИС и ее БД, позволяющей получать результаты обработки данных, хранящихся в БД ИС, в удобной для пользователей форме, но и основой средств разработки ИС – средств создания и изменения моделей ИС.

Первые две задачи решаются *средствами репортинга*, которые включаются в большинство ИС, но, в основном, их реализация такова, что для создания новых отчетов и запросов необходимо вмешательство программиста. Особенностью предлагаемого подхода является обеспечение интерфейса пользователя, облегчающего пользователям-непрограммистам подготовку отчетов, в том числе достаточно сложных [2].

Задача *поиска документов* в существующих информационно-поисковых системах решается в основном на основе концепции ключевых слов, т.е. без учета семантики.

В соответствии с предлагаемым подходом для описания семантики данных документа и его структуры используется *онтология* [3, 4]. В качестве подхода к решению описанной выше задачи был выбран онтологиче-

ский подход, в котором *онтология описывает как структуру, так и содержание документа*. Учитывая специфику решаемых в данной работе задач, будем считать, что онтология – это спецификация некоторой предметной области, которая включает в себя словарь терминов (понятий) предметной области и множество связей между ними, которые описывают, как эти термины соотносятся между собой в конкретной предметной области.

Для построения иерархии понятий онтологии используются следующие базовые типы отношений: «класс–подкласс», «часть–целое», «экземпляр–класс», «причина–следствие». Следует учесть, что данные типы отношений являются базовыми и не зависят от онтологии, но необходимо предоставить пользователю возможность добавления новых отношений, которые бы учитывали специфику описываемой предметной области. Кроме отношений любая онтология включает в себя два типа вершин. К первому типу отнесем вершины, описывающие структуру документа. Например: таблица, дата, должность и т.д. – они представляют собой общие понятия, не зависящие от конкретной предметной области. Другим типом будут являться вершины, содержащие понятия документа.

Фактически в данном контексте онтология – это *иерархическая понятийная основа рассматриваемой предметной области*. Онтология документа используется для анализа документа, благодаря ей из документа можно получить требуемые данные: известно, где искать данные и как они могут быть интерпретированы. Если представлять документ с использованием онтологий, то задача сопоставления онтологии и имеющегося документа сводится к задаче поиска понятий онтологии в документе. Как следствие, системе необходимо ответить на вопрос: описывает ли данная онтология документ или нет. На последний вопрос можно ответить утвердительно, если в процессе сопоставления в документе были найдены все понятия, включенные в онтологию.

Прежде, чем производить поиск вершин, содержащих понятия документа, необходимо провести поиск вершин, описывающих структуру документа. Таким образом, исходная задача сводится к задаче поиска в тексте документа общих понятий на основе формальных описаний.

Для решения проблемы выделения общих понятий на основе формальных описаний предлагается агентный подход [4]. Базу знаний агента для поиска общих понятий онтологии удобно представлять также в виде онтологии. Для предоставления пользователю возможности добавления новых шаблонов необходимо выделить базовые понятия для формирования общих.

Список литературы: 1. Lyadova L. The Multilevel Metadata as the Basis of Technology for Creation of Information Systems // Information Technologies and Telecommunications in Science and Education (IT&T ES'2005). Materials of the International Scientific Conference / Turkey, 2005. Moscow: VIZCOM, 2005. Pp. 83-86. 2. Ланин В. Архитектура и реализация средств репортинга в динамически настраиваемых информационных системах // Proceedings of the Fifth International Conference "Information Research and Applications" i.TECH 2007. V. 2. Varna, 2007. Pp. 348-356. 3. Lanin V., Lyadova L. Intelligent Search and Automatic Documents Classification and Cataloging Based on Ontology Approach // International Journal "Information Theories & Applications". V. 14/2007, Number 1. Pp. 25-29. 4. Ланин В. Интеллектуальное управление документами как основа технологии создания адаптируемых информационных систем // Сборник трудов международных научно-технических конференций «Интеллектуальные системы» (AIS'07) и «Интеллектуальные САПР» (CAD-2007). Научное издание в 4-х томах. Т. 2. М.: Физматлит, 2007. С. 350-357.

Поступила в редколлегию 18.02.08

УДК 512.086

Г.МАЙШМАЗ, НТУ «ХПИ»

ПРИНЯТИЕ РЕШЕНИЙ ПРИ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Розробка програмного забезпечення завжди пов'язана з прийняттям важливих керівницьких та інженерних рішень. Від цих рішень суттєво залежить якість програмного продукту та ризики, пов'язані з його розробкою. Пропонується автоматизувати процес прийняття рішень в процесі розробки програмного забезпечення з використанням формальних методів, що дозволить приймати рішення на основі об'єктивного аналізу фактів, чітко обґрунтувати зроблені висновки і оцінювати наслідки прийнятих рішень, й цим зводити до мінімуму ризики, пов'язані з розробкою програмного продукту.

Software development always relates to important management and engineering decisions. The quality of software and risks related to its development considerably depend on these decisions. It's proposed to automate the process of decision making during software development process using formal methods of decision making. This will allow to make decisions basing on objective analysis of facts, clear ground made conclusions and estimate consequences of the decisions, and minimize hereby risks related to development of software.

1. Введение. На данный момент существует множество программных продуктов, применяемых для поддержки и сопровождения различных этапов создания ПО: анализа системных требований, моделирования системы, ее отладки и тестирования. Данные программы являются CASE (Computer-Aided Software Engineering) средствами.

Все современные методы создания ПО используют соответствующие CASE-средства: редакторы нотаций, применяемые для описания моделей,

модули анализа, проверяющие соответствие модели правилам метода, и генераторы отчетов, помогающие при создании документации на разрабатываемое ПО. Кроме того, CASE-средства могут включать генератор кода, который автоматически генерирует исходный код программ на основе модели системы, а так же руководство пользователя. [1]

Сравнительная характеристика и возможности современных CASE-средств представлена в источнике [2].

Процесс разработки должен быть построен таким образом, чтобы обеспечить возможность измерения качества продукта. В практике программирования наиболее часто в роли метрики качества продукта выступает остаточная плотность ошибок, то есть плотность ошибок на тысячу строк кода или на одну функциональную точку. Однако если под качеством понимать степень удовлетворения требований, то мы должны измерять выполнение требований в конечном продукте.

В полной мере управлять качеством можно, если оно измеряется на всех этапах жизненного цикла. Качество к промежуточному продукту может быть установлено на основе отраслевых стандартов, в данном случае стандартов программирования (например, ISO или IEEE). [3]

CASE-средства позволяют автоматизировать многие рутинные операции на различных этапах разработки программного обеспечения, однако они слабо ориентированы на поддержку процесса принятия решений. Однако разработка ПО на разных этапах требует принятия сложных управленческих и инженерных решений. Это особенно важно для ранних стадий разработки ПО, т.к. ошибки на этом этапе связаны с серьезными рисками и могут привести к значительному увеличению затрат на разработку, или даже к краху проекта.

Эффективность выбора таких решений можно повысить за счет использования формальных методов — объективный анализ фактов, обоснование сделанных выводов и оценка последствий различных действий позволяют свести к минимуму риски, связанные с разработкой программного продукта. [4]

Управляющие проектами принимают множество решений в процессе разработки системы, которые влияют на итог проекта. Успешность проекта часто зависит от верности и точности этих решений. Исследования показывают, что управленческие факторы проекта обычно более критичны, чем поведенческие факторы, для успеха проекта. Принятие решений в процессе разработки особенно сложно в условиях не полной информации при наличии ресурсных и временных ограничений. Исследования показывают, что большинство сложностей возникают на ранних стадиях разработки.[5]