

УДК [004.4+004.031.6]:62.5

**СИСТЕМНЫЙ АНАЛИЗ ПРОЦЕССОВ РАЗРАБОТКИ, ИСПЫТАНИЙ
И ИСПОЛЬЗОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ***Статья поступила в редакцию 20.04.2017, в окончательном варианте – 17.06.2017.*

Брумштейн Юрий Моисеевич, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
кандидат технических наук, доцент, ORCID <http://orcid.org/0000-0002-0016-7295>,
e-mail: brum2003@mail.ru, https://elibrary.ru/author_profile.asp?authorid=280533

Куаншкалиев Тимур Ханатович, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
студент, ORCID <http://orcid.org/0000-0003-2732-1464>, e-mail: timurkyun@mail.ru, https://elibrary.ru/author_profile.asp?authorid=931136

Ильменский Михаил Александрович, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
студент, ORCID <http://orcid.org/0000-0001-8617-7242>, e-mail: ilmen---m@mail.ru,
https://elibrary.ru/author_profile.asp?id=931139

Колесников Иван Владимирович, Астраханский государственный университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 20а,
студент, ORCID <http://orcid.org/0000-0003-2584-3920>, e-mail: kiv.apple@gmail.com,
https://elibrary.ru/author_profile.asp?id=931138

Показано, что расширение использования робототехнических систем (РТС) в различных сферах деятельности приводит к необходимости повышения эффективности разработки, испытаний и применения программного обеспечения (ПО) для таких систем. Указаны общие и специфические характеристики ПО для РТС, в т.ч. связанные с особенностями его использования совместно с различными видами аппаратно-технических средств. Кратко охарактеризованы основные принципы обучения студентов/специалистов созданию ПО для РТС, управления качеством таких разработок, проверки/конт-роля характеристик (показателей) разработанного ПО. В жизненном цикле (ЖЦ) ПО для РТС авторами выделены следующие стадии (этапы): разработка ПО, включая его отладку; испытания разработанного ПО для РТС (автономные и в сочетании с аппаратными средствами систем); опытная и рабочая эксплуатация ПО; модернизация/доработка ПО с целью обеспечения продолжения его эксплуатации, в т.ч. с проведением (при необходимости) дополнительных испытаний; прекращение эксплуатации ПО. Рассмотрены типичные причины необходимости модернизации/доработки программного обеспечения для робототехнических систем, прекращения его эксплуатации. Для каждого из перечисленных выше этапов представлена совокупность подэтапов (подпроцессов, групп операций и т.д.), показаны их взаимосвязи в отношении обеспечения различных характеристик ПО для РТС. Проанализированы типичные виды угроз для процессов разработки, испытаний и эксплуатации ПО, в т.ч. в отношении неполноты обеспечения его функциональности, нарушений информационной безопасности и пр. Показана целесообразность использования при разработке, испытаниях и эксплуатации ПО методологии «управления проектами», методов риск-менеджмента (РМ). Для выделенных этапов и подэтапов ЖЦ ПО проанализированы целесообразные меры РМ, позволяющие снизить вероятности реализации существующих и возможных в будущем видов неблагоприятных событий, величин ущербов в случае реализации таких событий. Рассмотрена постановка и методы решения оптимизационной задачи по распределению «усилий» разработчиков между различными этапами ЖЦ ПО для РТС.

Ключевые слова: робототехнические системы, программное обеспечение, разработка, испытания, опытная эксплуатация, рабочая эксплуатация, модернизация, прекращение эксплуатации, распределение ресурсов, методы оптимизации, информационная безопасность, структура рисков, управление рисками

**SYSTEM ANALYSIS OF DEVELOPMENT, TESTING AND USE PROCESSES
FOR THE ROBOTIC SYSTEMS SOFTWARE***The article has been received by editorial board 20.04.2017, in the final version – 17.06.2017.*

Brumshhteyn Yuriy M., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation

Cand. Sci. (Engineering.), Associate Professor, ORCID <http://orcid.org/0000-0002-0016-7295>,
e-mail: brum2003@mail.ru, https://elibrary.ru/author_profile.asp?authorid=280533

Kuanshkaliev Timur Kh., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation
student, ORCID <http://orcid.org/0000-0003-2732-1464>, e-mail: timurkyun@mail.ru, https://elibrary.ru/author_profile.asp?authorid=931136

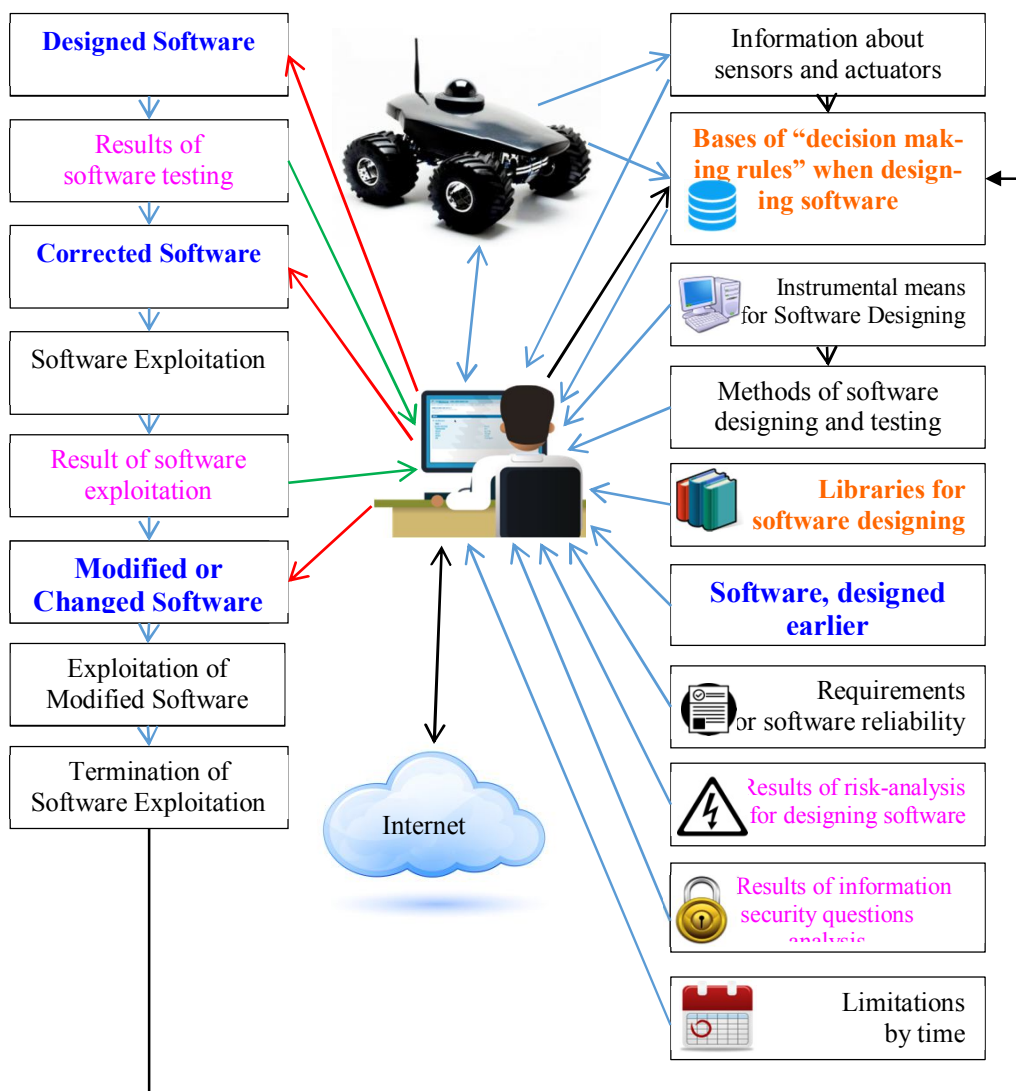
Ilmenskiy Mikhail A., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation
student, ORCID <http://orcid.org/0000-0001-8617-7242>, e-mail: ilmen---m@mail.ru,
https://elibrary.ru/author_profile.asp?id=931139

Kolesnikov Ivan V., Astrakhan State University, 20a Tatishchev St., Astrakhan, 414056, Russian Federation
 student, ORCID <http://orcid.org/0000-0003-2584-3920>, e-mail: kiv.apple@gmail.com,
https://elibrary.ru/author_profile.asp?id=931138

The expanding use of robotic systems (RS) in different spheres requires efficiency increasing for development, testing and implementation of software (SW), intended for such systems. The paper identifies general and specific characteristics of SW for RS, including those coupled with features of compatible use with different types of hardware. The authors also give a short outline of the main principles of training students/experts to develop SW for RS, carry out quality management of such development, check and monitor the developed SW characteristics. In the life cycle (LC) of SW for RS authors have allocated the following stages: SW development, including its debugging; testing of the developed SW for RS, including independent tests and tests of SW work with RS hardware; the SW trial and working exploitation; the SW upgrade/modification to provide operation continuation, including (if necessary) additional tests; termination of SW exploitation. The typical reasons for the SW upgrade/modification, as well as its termination of exploitation, are considered. The paper describes sub-phases (sub-processes, groups of operations etc.) for each of the phases listed above; shows interrelations between them, ensuring the support of SW for RS different characteristics. The authors also analyze the typical kinds of threats for the processes of development, testing and use of SW for RS, including incomplete system functionality, information security violations etc. In the article are proved relevance of project management methodology and risk management (RM) methods while developing, testing and using the SW for RS. Also are analyzed efficient measures of RM, allowing to reduce probable unfavorable events at present and in the future, as well as to decrease losses in case these events do occur for all the defined phases and sub-phases of the software LC. The authors deal with formulation and ways to solve of optimization problem, concerned with developers efforts distribution between different LC phases of software for RS.

Keywords: robotic systems, software, development, tests, trial operation, working operation, upgrade, operation extinction, distribution of resources, optimization methods, information security, structure of risks, risk management

Graphical annotation (Графическая аннотация)



В настоящее время расширяется использование робототехнических систем (РТС) в различных сферах деятельности, включая оборонный комплекс страны, производство продукции гражданского назначения, сферу услуг, образование, здравоохранение и пр. Эти процессы сопровождаются опережающей регистрацией большого количества патентов, что дает возможность оценить тенденции и долгосрочные перспективы развития робототехнической отрасли, спрогнозировать потребности в специалистах [15]. Используемое в РТС программное обеспечение (ПО) становится все более сложным и «интеллектуальным» (кроме РТС чисто учебного назначения, разрабатываемых студентами). Процессы разработки, испытаний и использования (РИИИ) ПО для РТС имеют значительную специфику, которая в существующей литературе отражена относительно слабо. Поэтому целью настоящей статьи является системный анализ процессов РИИИ ПО для РТС. При этом акценты делаются на такие вопросы: риск-менеджмент процессов РИИИ ПО; анализ и управление уровнями информационной безопасности (ИБ) ПО для РТС на всех стадиях его жизненного цикла (ЖЦ) [4], в т.ч. в рамках подходов «управляемых моделями» [38].

Общая характеристика процессов разработки и использования ПО для РТС. С точки зрения разработки ПО целесообразно различать мобильные РТС [1, 2, 17, 18, 26, 29] и немобильные. Первый вариант обычно сложнее в реализации, в т.ч. из-за необходимости обеспечения устойчивости РТС при передвижении (особенно при малом количестве опор) [1, 26].

Возможны различные варианты «компонентной структуры» ПО для РТС.

1. Единственная программа, разработанная на том или ином языке программирования (ЯП) или совокупности ЯП. Эта программа может быть предназначена для использования на РТС учебного характера; на уникальном или серийно выпускаемом РТС; на РТС, предназначенной для решения исследовательских задач. Для размещения разработанного ПО могут использоваться различные аппаратные средства, включая микропроцессоры и одноплатные компьютеры [19].

2. Комплекс взаимосвязанных программ, которые обеспечивают выполнение различных функций, необходимых для эксплуатации РТС. Такие программы могут разрабатываться параллельно, что позволит сократить общие сроки разработки. В составе комплексов ПО в зависимости от их архитектуры [24] могут применяться различные библиотеки прикладных программ, предназначенные для обслуживания устройств, решения вычислительных задач и пр.

3. Программа или комплекс программ в сочетании с одной или *большим* числом баз данных (БД) и / или баз знаний, в т.ч. сформированных по результатам эксплуатации РТС в автоматическом или ручном режимах.

При разработке ПО выбор из перечисленных выше вариантов может осуществляться с учетом следующего: характера задач, решение которых должно обеспечивать разрабатываемое ПО; имеющегося опыта разработчиков; наличия и степени освоенности ими инструментальных средств разработки (ИСР) ПО для РТС, включая архитектуру ПО [24]; наличия в распоряжении разработчиков аналогичного ПО, которое можно взять за прототип (основу); предельно допустимых сроков создания ПО; соображений ИБ, в т.ч. в отношении заимствуемого ПО или его фрагментов и др. Такой выбор является задачей оптимизации обычно при нечетких критериях оптимальности. При этом ограничения могут носить как барьерный (жесткий), так и мягкий характер. Математическая постановка задачи оптимизации обычно осложняется следующим причинами: наличием нескольких частных критериев оптимальности, относительная важность которых может быть оценена лишь нечетко; зависимостью набора таких критериев от характера задач; нечеткостью условий эксплуатации создаваемых РТС.

Разработка ПО для РТС может осуществляться с разными целями:

1. Учебными, в т.ч. при обучении в вузах студентов и магистрантов; в рамках занятий со специалистами на курсах повышения квалификации; при работе со школьниками / детьми (в т.ч. когда такое обучение осуществляют студенты [11] и пр.). Основные требования к создаваемому ПО: простота разработки (написания и отладки текстов программ); использование интуитивно понятных алгоритмических решений; получение работоспособного ПО с минимальными трудозатратами и в максимально короткие сроки; обеспечение наглядности получаемых результатов разработки ПО в виде РТС, выполняющих предусмотренные для них действия.

Для варианта 1 вопросы вычислительной эффективности такого ПО, надежности и ИБ его эксплуатации [35] носят неосновной характер. При разработке могут использоваться языки визуального программирования [39]. Они не предъявляют высоких требований к квалификации разработчиков и делают сам процесс создания ПО простым и наглядным.

2. Разработки ПО для РТС, носящие исследовательский (поисковый) характер. Типичная цель – сравнение эффективности различных вариантов программно-технических решений, в т.ч. в сочетании с аппаратно-техническими (включая телекоммуникационные средства). Как следствие, для варианта 2 этапы разработки и испытаний ПО могут быть достаточно длительными по времени; по результатам испытаний могут неоднократно проводиться корректировки ПО для повышения его качества [35], надежности, уровня ИБ. Требования к простоте разработки ПО обычно не выдвигаются. Однако могут быть особенно важны вопросы обеспечения полноты проверки всех «логических ветвей» ПО при планировании и проведении испытаний.

В варианте 2 выделим разработки ПО для РТС, предназначенные для исследования (оценки) защищенности их от угроз ИБ. При этом важны угрозы, направленные не только на ПО, но и на телекоммуникационные средства, на каналы связи РТС с центрами управления (ЦУ), а также РТС друг с другом при их действиях в составе групп.

Расширение использования антропоморфных роботов (АМР) делает актуальной задачу совершенствования управления их мимическим аппаратом. Разработка и отладка соответствующего ПО [20] представляет особую сложность. Основные причины: необходимость использования согласованного по времени управления большим количеством сервоприводов; требования сочетаемости мимики АМР с воспроизводимой ими речью; необходимость обеспечения адекватных мимических реакций АМР на внешние вербальные (а иногда и тактильные) воздействия; требования соответствия мимики с жестами роботов, с передвижениями / поворотами их корпусов и пр.; экспертный (субъективный) характера оценок правдоподобности воспроизводимой АМР мимики.

3. Создание ПО для РТС, предназначенных для рабочей эксплуатации и / или серийного выпуска. В этом случае на первом плане находятся такие вопросы: снижения трудоемкости и сроков разработки ПО за счет использования ИСР [24]; качества [35] разработанного ПО; эксплуатационной надежности ПО; его вычислительной эффективности; ИБ ПО. Как следствие, большое внимание уделяется вопросам управления качеством создаваемого ПО, проведению его испытаний: автономных и в составе РТС. Последние обычно можно считать сложными техническими системами (СТС) и применять подходы, разработанные для испытаний СТС [10, 31, 32].

Важнейшими особенностями ПО для РТС являются следующие. (1) Эксплуатация ПО в режиме реального времени. (2) Использование информации с сенсорных систем (СС), систем технического зрения (СТЗ). (3) Применение ПО для управления исполнительными механизмами. При этом необходимо учитывать следующее: дискретность во времени получения информации, используемой для управления, с СС и СТЗ РТС, из ЦУ и из иных источников; конечные продолжительности во времени операций, выполняемых аппаратными средствами РТС, включая манипуляторы; ограниченные скорости перемещения/поворотов РТС в пространстве; ограничения по углам поворотов конечностей РТС, размерам преодолеваемых препятствий и пр.; наличие некоторых минимальных смещений для подвижных частей РТС, люфтов в механических передачах и пр. Во многих случаях в ПО для РТС используется достаточно сложная логика выполнения операций; может применяться обучение или самообучение ПО для РТС с целью адаптации к изменяющимся внешним условиям, номенклатуре угроз ИБ и пр.

В качестве СС РТС в настоящее время используются не только различные типы датчиков (в т.ч. комбинированных), но и системы ультразвуковой и электромагнитной локации, реже – тактильные датчики. Навигационные системы мобильных РТС могут носить инерциальный характер, использовать данные о пройденной траектории, применять сведения о расположении РТС по отношению к некоторым реперным объектам с известными координатами. В составе РТС могут использоваться СТЗ видимого и инфракрасного диапазона; библиотеки ПО, предназначенные для распознавания образов на изображениях, получаемых с СТЗ [25] и др.

На центры ЦУ РТС (внутренние или внешние) с СС и СТЗ поступают большие объемы информации, которые необходимо обрабатывать в режиме реального времени. Это выдвигает серьезные требования к быстродействию вычислительных систем (ВС) РТС; к эффективности алгоритмов, используемых в ПО.

Перечислим основные причины ограниченности скоростей вычислений, объемов оперативной и энергонезависимой памяти ВС, находящихся в составе РТС. 1. Массогабаритные ограничения на конструкции мобильных РТС. 2. Ограничения по энергопотреблению СС, СТЗ, ВС в мобильных РТС с автономными источниками питания. 3. Если эксплуатация РТС предполагается в условиях значительных радиационных воздействий, то это требует применения специальных стойких к радиации микросхем, имеющих относительно невысокие вычислительные возможности. 4. Эксплуатация РТС в условиях высоких уровней вибраций и / или при больших перепадах температур требует дополнительного расхода масс конструкций и энергозатрат для обеспечения приемлемых условий работы ВС, СС, СТЗ и пр.

Ограничения на возможности ВС, работающих в составе РТС (по указанным выше причинам), влияют и на подходы / решения, используемые при разработке ПО.

Возможные варианты управления РТС.

1. Автономное – на основе использования ПО, установленного на микропроцессоре (МП), микроконтроллере (МК), реже – на микроЭВМ или однокристальных компьютерах [19]. При этом ПО может обеспечить автономную работу РТС по заданным алгоритмам без использования внешних по отношению к ним ЦУ. Важными подвариантами этого варианта можно считать разработку ПО для самоадаптирующихся [36] и самореконфигурируемых [40] РТС.

2. Подключение РТС к ЦУ (в которых используются компьютеры или их комплексы) по беспроводным или проводным каналам связи для обеспечения управления в автоматическом режиме или в рамках человеко-машинных АСУ [34]. При этом необходимо учитывать специфику взаимодействия ПО, размещенного на РТС и на внешних для них компьютерах [27].

Проводные каналы связи с ЦУ обычно являются более стабильными в работе и более устойчивыми к угрозам ИБ. Однако они резко ограничивают мобильность РТС.

Беспроводные каналы связи (БКС) обычно имеют меньшую (по сравнению с проводными) надежность; часто – переменную скорость передачи данных; изменяющиеся величины задержек поступления информации на ЦУ и на РТС. Для защиты БКС от несанкционированного перехвата информации может использоваться шифрование данных. Если для обеспечения работы БКС с РТС используется Интернет и / или сети операторов сотовой связи, то появляются дополнительные угрозы ИБ.

Вариант с ЦУ дает возможность применять значительно более мощное компьютерное оборудование, но он имеет следующие недостатки. (А) Делает РТС более уязвимыми к угрозам, которые могут воздействовать на «каналы связи» (включая помехи); к угрозам, направленным на ЦУ; к ошибкам людей-операторов в виде их не адекватных и / или несвоевременных действий [34]. (Б) Вносит дополнительные задержки во времена реакций РТС на изменения окружающей среды; на изменения ситуаций; на появление угроз для РТС, в т.ч. угроз ИБ.

3. Гибридный вариант – основную работу выполняет ПО, размещенное на РТС, а управление со стороны ЦУ осуществляется только периодически и / или в тех случаях, когда необходимы решения, не предусмотренные в ПО, установленном на РТС.

Для варианта 1 относительная простота используемого ПО, отсутствие сложных операционных систем потенциально снижают угрозы несанкционированного внедрения вредоносного ПО. В качестве средств защиты от такого внедрения могут использоваться следующие приемы: при запуске исполнимой программы на МП или МК сравнивается циклическая контрольная сумма (CRC) с хранимым в самом ПО эталонным значением (самотестирование ПО); сравнение CRC для файлов с принятыми командами управления РТС (однако включение CRC в такие файлы увеличивает их размеры). Для вариантов 2 и 3 на ПЭВМ внешнего (по отношению к РТС) ЦУ может использоваться антивирусное ПО, которое распознает и т.н. промышленные вирусы.

В случае групп РТС возможны такие схемы организации управления.

1. Самоуправление группы РТС без использования ЦУ. При этом РТС-лидеры, осуществляющие координацию действий РТС в группах, могут со временем меняться.

2. Согласованное управление группой РТС из ЦУ. При этом может использоваться информация как с СС и СТЗ РТС, так и из других источников, включая средства видеонаблюдения, радиотехнические системы сканирования каналов связи и пр. Если для БКС обнаруживается высокий уровень помех, то в ЦУ и на РТС может осуществляться автоматическое переключение на резервные каналы. При потере связи с ЦУ для РТС предполагается переключение на резервный алгоритм действий, например, на остановку перемещения мобильной РТС или ее частичный возврат по пройденной ранее траектории. Для снижения уязвимости БКС к помехам возможно и аperiodическое переключение между выбранным набором каналов по заданному алгоритму.

Если для всех РТС используется общий канал связи, то необходимо включение в отправляемые ЦУ сообщения кодов РТС-получателей; в сообщения, отправляемые РТС, должны включаться коды РТС-отправителей.

3. Гибридный вариант по отношению к 1 и 2, при котором ЦУ задействуется только в сложных случаях и/или применяется для информационного обмена между РТС.

Для обеспечения ИБ связи ЦУ с РТС может применяться шифрование информации, передаваемой в обоих направлениях, с использованием таких решений. (А) Единый закрытый ключ для шифрования-дешифрования информации по всем РТС. При этом в состав передаваемой с ЦУ на РТС информации необходимо включать закодированный адрес РТС получателя, если передача осуществляется не по узко-направленному радио- или лазерному лучу конкретной РТС. Аналогично в состав информации, передаваемой с РТС необходимо включать его код (при тех же условиях). Такое решение в принципе позволяет пользоваться общим радиоканалом для всех РТС, в случае, если информация с РТС отправляется по запросам с ЦУ или по определенному графику, разному для каждой из РТС (для исключения одновременной передачи данных с разных РТС на ЦУ). (Б) Используются индивидуальные закрытые ключи для связи ЦУ с каждой из РТС. (В) Применение несимметричных схем шифрования с открытыми ключами, аperiodически рассылаемыми ЦУ на РТС. Представляется, что в этом случае увеличивается вероятность подмены (имитации) информации, передаваемой с конкретных РТС в ЦУ, но только в том случае, если злоумышленнику известна структура передаваемой с РТС информации и способ кодирования номера РТС, передающей информацию. В то же время перехват информации с РТС при использовании «открытых ключей» не позволяет определить содержание информации. Однако если при перехвате удастся определить источник сигнала, то это может «демаскировать» передающую РТС, а также фиксировать сам факт передачи с нее информации при определенных условиях. Последнее иногда может быть важным для определения «логики управления» РТС.

Дополнительными средствами защиты информации при приеме-передаче могут быть смена по специальной временной схеме частотных каналов, применяемых для связи ЦУ с РТС; использование разных пар частот для связи ЦУ->РТС и РТС->ЦУ и др.

Для организации связи между группами самоуправляющихся РТС необходима однозначная идентификация РТС-отправителей и РТС-получателей сообщений. Она может осуществляться за счет включения в сообщения соответствующих кодов-идентификаторов, закрепленных за каждой из РТС. Для шифрования информации, передаваемой между РТС, могут использоваться два варианта: закрытые ключи для шифрования-дешифрования (в т.ч. индивидуальные ключи для каждого сочетания РТС в парах); открытые ключи, периодически рассылаемые каждой из РТС другим РТС, входящим в группу.

В разработках РТС сейчас характерно применение готовых МК, обеспечивающих необходимые сервисные функции по взаимодействию с внешними для них устройствами на основе общедоступных фрагментов кодов ПО для обеспечения такого взаимодействия. Дистанционное обновление ПО в МК по ходу использования РТС возможно, но это потенциально увеличивает риски ИБ, в т.ч. в отношении подмены ПО.

Выбор МП (МК) для РТС определяется такими аппаратными характеристиками: требованиями к количеству входных / выходных каналов для получения / передачи информации; наличием и количеством встроенных в МП аналогово-цифровых преобразователей (АЦП) и / или цифро-аналоговых преобразователей (ЦАП); допустимыми для нормальной эксплуатации МП характеристиками (температура, вибрационные нагрузки, уровень радиации и пр.); быстродействием МП. При недостаточности в МП количества входных каналов могут быть использованы мультиплексоры. Однако технически это сложнее в реализации, чем выбор иного МП или МК. Отметим еще, что использование в РТС МП или МК зарубежного производства приводит к увеличению рисков ИБ, связанных с возможным наличием в таких устройствах «недокументированных команд».

При разработке ПО для РТС наиболее важны следующие характеристики МП: тактовая частота МП; его разрядность; объем памяти (особенно при большом по размеру ПО и / или БД); набор допустимых ЯП и / или ИСР [24], их функциональные возможности и удобство создания кода; наличие в ИСР отладочных возможностей в отношении ПО, включая пошаговую отладку программ и имитацию (эмуляцию) работы внешних устройств, вывод значений переменных на отдельные экраны; наличие готовых библиотек ПО для работы с внешними устройствами, включая дисплеи различных типов, СТЗ и пр. С позиций разработчиков РТС при выборе МК обязательно учитывается наличие в них необходимых аппаратно-программных средств для обеспечения связи с внешними устройствами. Это позволяет минимизировать работу по созданию принципиальных электрических схем и их изготовлению, уменьшает объемы работы по созданию ПО.

В существующих публикациях обычно не рассматриваются вопросы утечек информации по электромагнитному каналу (ЭМК) с электрических схем, включающих МП (в т.ч. используемых в РТС). Основная причина – размеры соединительных проводников, которые могут играть роль излучающих антенн, в таких схемах малы. Невелика и интенсивность излучения. Однако потенциально перехват информации с таких схем по ЭМК возможен при следующих условиях: отсутствия металлических кожухов, экранирующих электромагнитное излучение; применения чувствительных сканирующих приемников нужного диапазона, расположенных близко от МП. Кроме того, отсутствие таких кожухов (экранов) увеличивает уязвимость электрических схем к мощным электромагнитным импульсным воздействиям, специально предназначенным для выведения таких схем из строя.

При анализе необходимости разработки нового ПО для РТС неучебного характера целесообразно прежде всего выяснить, нет ли уже соответствующего готового ПО (особенно при решении типовых задач). В случае ПО, предназначенного для ПЭВМ, поиск программ-аналогов возможен в различных источниках, в т.ч. с применением поисковых систем Интернета; специализированных поисковых систем научных электронных библиотек; поисковых средств специализированных сайтов по математическим алгоритмам, зарегистрированным программам для ЭВМ и пр. В то же время в отношении ПО для РТС в России специализированные сайты фактически отсутствуют, хотя они есть за рубежом.

На сайте Федерального института промышленной собственности (ФИПС) в поисковой системе для зарегистрированного ПО нет поля фильтра, позволяющего выделить программы, предназначенные специально для МП или МК. Однако отбор в базе ФИПСа ПО для МП и МК (или непосредственно для РТС) возможен с использованием соответствующих ключевых слов, которые могут встречаться в рефератах к зарегистрированному ПО. Анализ показал, что специализированного ПО, предназначенного для РТС, там относительно немного. При этом в действующих правилах ФИПСа нет ограничений в отношении возможностей регистрации ПО для РТС. С точки зрения авторского права предназначение ПО также не играет роли в отношении возможности его регистрации в виде программы для ЭВМ и / или БД [8].

Отметим, что в ИСР [24] для ПО, предназначенного для использования на МП и / или МК, обычно есть примеры программного кода для решения типовых задач. Это касается, в частности, взаимодействия МП / МК с различными датчиками, исполнительными механизмами, устройствами индикации и пр.

Основные особенности создания кодов ПО для РТС русскоязычными разработчиками – необходимость использования иноязычных ИСР и библиотек, т.к. для них часто нет русификаций; применение

комментариев в кодах программ на русском языке приводит к необходимости частых переключений раскладок клавиатуры; использование в кодах программ мнемонических обозначений переменных на английском языке требует достаточно хорошего владения этим языком; нецелесообразность одновременного использования англоязычных и транслитерированных на латиницу русскоязычных обозначений для переменных в кодах программ; необходимость понимания англоязычных комментариев в примерах листингов программ, включенных в ИСР.

В заключение раздела отметим, что тексты программ для ЭВМ в рамках авторского права охраняются как литературные произведения (пункт 1 ст. 1259 ГК РФ). В то же время сами РТС в целом (включая ПО для них) в соответствии с главой 77 ГК РФ могут рассматриваться как объекты единых технологий [6, 7].

Этап собственно создания ПО для РТС. В отношении предназначения разрабатываемого ПО целесообразно выделить такие его типы. (А) Для отдельных РТС, работающих автономно, т.е. у них нет информационных связей по управлению с ЦУ или другими РТС. (Б) Для отдельных РТС, управляемых с внешнего ЦУ. При этом такое управление может носить как полный характер, так и частичный, т.е. в сочетании с использованием управляющего ПО на самом РТС. (В) Для групп РТС, согласованно управляемых внешним ЦУ на основании получаемой с них оперативной информации. (Г) Для групп РТС, взаимодействующих друг с другом по данным и управлению (без использования внешнего ЦУ). (Д) Для групп РТС, взаимодействующих друг с другом непосредственно и, кроме того, дополнительно управляемых из ЦУ. При групповом самоуправлении РТС в порядке самоорганизации систем могут формироваться (в т.ч. и динамически) некоторые иерархии РТС. Функциональные различия между отдельными РТС могут определять их принадлежность к некоторым группам устройств в гетерогенных [37] системах.

Таким образом, процессы создания и проведения испытаний ПО для РТС могут требовать разработки взаимно согласованного ПО как при использовании ЦУ, так и без него. Разработка, отладка, испытания для совокупности ПО (или даже единого ПО для использования на взаимодействующих РТС) сложнее, чем для единственной автономной РТС. Это касается также вопросов обеспечения ИБ для ПО.

Описываемые ниже одиннадцать подэтапов для этапа разработки (создания) ПО обычно выполняются в приведенной последовательности. Однако при выявлении нештатных ситуаций (прежде всего ошибок и недочетов в ПО) этот порядок может нарушаться из-за необходимости возврата к более ранним подэтапам. Такой возврат может происходить и с этапа проведения испытаний ПО, который рассматривается в последующем разделе.

1. Начальный (первый) подэтап этапа разработки ПО – определение его функциональности. Она вытекает из задач, которые должна решать РТС. При этом иногда приходится учитывать и медико-технические требования [29], специфические требования в отношении условий внешней среды [33] и др.

В некоторых случаях может быть более рациональной разработка типового ПО с расширенной функциональностью (для применения на различных РТС), вместо индивидуальных разработок ПО для каждой из них. Разработка типового ПО может быть предпочтительной по соображениям стоимости и надежности (провести полноценные испытания проще для одной системы управления РТС, чем для нескольких).

2. Определение состава входных данных для работы ПО (в т.ч. получаемых в автоматическом режиме в реальном времени), точности этих данных, периодичности их съема с СС и СТЗ и пр. Частота съема может управляться динамически [9] с учетом внешней для РТС обстановки и скорости ее изменения. Это позволяет реализовать и спящие режимы автономных РТС для снижения энергозатрат в режиме сна.

3. Выбор алгоритмов из числа существующих (известных, описанных в литературе) или разработка новых (если подходящих аналогов найти не удалось) [5]. Сами по себе алгоритмы не являются объектами авторского права [8]. Однако способы, которые реализуют эти алгоритмы, могут быть отражены в действующих или уже прекративших действие патентах на изобретения или полезные модели. Алгоритмы могут относиться не только к решению вычислительных и логических задач, но и работе с различными внешними для РТС устройствами.

Если по результатам патентного анализа будет выявлено, что необходимый для использования при управлении РТС способ (соответствующий алгоритму) соответствует действующему патенту, то юридически корректными могут быть два вида решений: 1. Получение неисключительной лицензии у правообладателя на право использования запатентованного способа. 2. Самостоятельная разработка иного способа (и реализующего его алгоритма) для обеспечения патентной чистоты разработки ПО.

При достаточно сложных алгоритмах для их анализа и проверки логических ветвей может быть целесообразным (особенно в учебных целях) построение блок-схем [21]. При этом иногда рационально создать укрупненную блок-схему, а затем для каждого из укрупненных блоков построить отдельные (частные) блок-схемы.

Если в рамках создания ПО для РТС выполнена разработка нового способа (алгоритма), то он потенциально может быть зарегистрирован как объект промышленной собственности. Однако большие трудовые и финансовые затраты, связанные с регистрацией, достаточно часто делают ее нерациональной (особенно, если ПО, в котором будут использоваться алгоритмы, не предназначено для серийного выпуска).

4. Оценка возможности повторного использования ПО, ранее созданного теми же разработчиками. Такие возможности могут ограничиваться договорами с заказчиками, в интересах которых было

ранее создано ПО для РТС. Однако в большинстве случаев прямые запреты на повторное использование фрагментов ПО для иных систем, можно обойти за счет корректировок программного кода.

Целесообразность повторного использования ПО или его фрагментов (в т.ч. и с некоторыми переработками), может определяться с учетом результатов поиска информации по пункту 5. Однако если уже имеющееся ПО обладает всей необходимой функциональностью, то этап 5 может быть вообще не нужным.

5. Поиск в доступных источниках информации об уже существующем ПО с необходимой или более обширной функциональностью, по которому представлен исходный код и / или это ПО имеет открытую архитектуру. Здесь под ПО понимаются не только основные программы, но и библиотеки подпрограмм.

Если нужное ПО будет найдено, то возможны следующие варианты. (А) Использование этого ПО без каких либо дополнительных мер, если оно распространяется бесплатно (например, на основе лицензии типа "Creative Common" или BSD-лицензии). (Б) В ином случае юридически корректное действие – приобретение права на использование ПО (неисключительной лицензии) у правообладателей (если условия получения такого права являются приемлемыми для разработчиков).

6. Если при поиске удалось найти тексты лишь примерных аналогов необходимого ПО, то вместо разработки с нуля может быть целесообразным выбор наиболее подходящего аналога и модификация (корректировка) существующего кода, написанного на одном из высокоуровневых ЯП. Возможности модификации (корректировки) кода могут ограничиваться следующими факторами: отсутствием в исходном коде комментариев хотя бы к ключевым блокам (операторам) программы; нарушением в исходном коде правил структурного представления текстов программ; запутанной логикой исходного кода. Отметим, что иногда разработчики ПО умышленно делают его логику малопонятной для других лиц.

Согласно статье 1280 ГК РФ для лиц, правомерно владеющих *«экземпляр программы для ЭВМ или экземпляром базы данных ... без разрешения автора или иного правообладателя и без выплаты дополнительного вознаграждения»*, допустимы следующие действия. По подпункту 1 п. 1 ст. 1280 *«внесение в программу для ЭВМ или базу данных изменений исключительно в целях их функционирования на технических средствах пользователя, исправление явных ошибок, если иное не предусмотрено договором с правообладателем»*. По пункту 2 возможность *«...изучать, исследовать или испытывать функционирование такой программы в целях определения идей и принципов, лежащих в основе любого элемента программы для ЭВМ»*. По пункту 3 при соблюдении ряда условий, перечисленных в подпунктах 1–3 к этому пункту, *«...преобразовать объектный код в исходный текст (декомпилировать программу для ЭВМ) или поручить иным лицам осуществить эти действия, если они необходимы для достижения способности к взаимодействию независимо разработанной этим лицом программы для ЭВМ с другими программами, которые могут взаимодействовать с декомпилируемой программой»*.

Иногда может быть целесообразным исключение (удаление) разработчиками (или даже эксплуатантами) излишней функциональности из уже существующего ПО для уменьшения размера программы, увеличения ее быстродействия и пр. Однако корректировка текста программы для ЭВМ может нарушать личные неимущественные права ее разработчиков [8], в т.ч. и при бесплатном распространении ПО.

Даже при наличии у разработчиков текстов программ-аналогов иногда может быть более целесообразным написание ПО с нуля, т.к. разобраться в логике чужой программы часто сложнее, чем написать собственную. Кроме того, заимствование для разработок чужого ПО (или даже его фрагментов), использование открытого ПО увеличивает риски ИБ, в т.ч. из-за наличия в нем недокументированных возможностей.

Даже если этап самостоятельной разработки ПО оказался не нужен, то обычно необходим этап испытаний заимствованного ПО в составе конкретной РТС.

7. Если необходимые средства получения информации (СС, СТЗ, локационные системы) и исполнительные механизмы уже были выбраны при конструировании РТС, то обычно необходим выбор / подбор МП или МК с учетом факторов, указанных в предыдущем разделе. Важное значение может играть и опыт работы создателя ПО с МП или МК в рамках выполнения предшествующих разработок.

Оптимальный выбор МП или МК (из числа допустимых) может осуществляться на основе интегрального критерия (ИК), учитывающего значения всех факторов с разными весовыми коэффициентами (т.е. фактически аддитивной свертки информации о технико-экономических характеристиках устройства). На практике такие ИК трудноприменимы, т.к. выбор весовых коэффициентов может быть весьма индивидуальным для каждого лица, принимающего решения. Кроме того, учет совместимости выбираемых устройств фактически приводит к задачам группового выбора вместо индивидуального (в т.ч. и за счет ограничений на суммарные расходы по всем устройствам). Описанные причины значительно усложняют модели принятия решений, делают обычные формулы для ИК в виде взвешенных усреднений по параметрам малоприменимыми.

8. Структурное проектирование ПО, включая определение номенклатуры функциональных блоков (и, возможно, подпрограмм, библиотек), взаимодействия их друг с другом по управлению и данным.

9. Создание (разработка, кодирование) исходных текстов ПО для РТС, в т.ч. с применением для этой цели ИСР [24] и специализированных библиотек подпрограмм [24, 25]. На этапе кодирования программ (а также выбора структуры ПО) большое значение уделяется вопросам обеспечения надежности программного обеспечения [3, 15, 35]. Использование готовых подпрограмм в рамках разработки ПО

возможно в двух вариантах: 1) в виде исходных текстов (кодов), компилируемых совместно с основными модулями программ; 2) объектных кодов, которые линкуются (связываются) с основными модулями на этапе сборки программы для получения файлов *exe*, *com*.

Для подэтапа 9 организация программирования возможна как по схеме сверху вниз (с последовательной детализацией выбранных алгоритмов с помощью блоков, а потом и операторов), так и снизу вверх (разработка начинается с отдельных блоков, которые затем компонируются в общую программу).

Существуют диалекты ЯП и компиляторы, специально предназначенные для создания программ для МП. Для уменьшения количества ошибок на этапе разработки в них предусмотрены средства контроля синтаксиса разрабатываемых программ, включая выделение различными цветами опознаваемых ключевых слов ЯП; использование средств автоматического форматирования текстов программ. Это позволяет выявить вложенность групп операторов в тексте кода программы. Однако специальные отладочные компиляторы, которые позволяют максимально полно выявить ошибки логики программы, как правило, не используются.

При написании кода ПО для РТС целесообразно применять следующие приемы:

- указывать в виде комментария в начале программы не только ее назначение, но и модели (характеристики) входящих в состав РТС устройств, с которыми ПО должно взаимодействовать;
- соблюдать правила структурного программирования при записи кода;
- выделять блоки ПО, предназначенные для решения отдельных вычислительных и логических задач, работы с внешними устройствами и пр.;
- использовать мнемонические имена для переменных и констант;
- включать в текст ПО хотя бы комментарии к блокам и операторам, в т.ч. в отношении взаимодействия с внешними устройствами, другими РТС, ЦУ;
- ограничивать использование операторов *goto*.

Помимо повышения надежности ПО эти меры облегчают и возможности его модернизации в последующий период.

При одновременной разработке ПО для групп РТС, содержащих функционально различные устройства, особое внимание должно уделяться обеспечению взаимодействия таких программ, особенно если их кодирование осуществляют разные программисты. При таких условиях целесообразно предварительная разработка документа типа соглашения о связях между модулями ПО. Это позволяет осуществлять разработку ПО параллельно, применяя методологию управления проектами.

Отдельно укажем на целесообразность разработки и использования блоков ПО, предназначенных для самотестирования РТС (при включении, в процессе их эксплуатации и пр.). При этом должно проверяться следующее: неизменность файла с ПО; каналы взаимодействия РТС с внешними устройствами. Последнее может быть важно для РТС, предназначенных для работы в неблагоприятных внешних условиях. Данные условия могут приводить к выходу из строя частей РТС. В этом случае может потребоваться перенастройка ПО, учитывающая возможности использования работоспособных частей. Для проведения самотестирования РТС могут использоваться дополнительные датчики; контроль подвижности и / или предельных перемещений манипуляторов РТС и пр.

При разработке ПО для групп РТС (или для РТС, взаимодействующих с ЦУ) должна проверяться также доступность каналов связи, в т.ч. при передаче информации во всех необходимых направлениях.

В большинстве существующих работ по созданию ПО для групп РТС (например, [12, 13]) считается, что численности РТС в группах фиксированы. Однако в общем случае это может быть не так: часть РТС может выходить из строя в процессе выполнения групповых задач, в т.ч. из-за получения механических повреждений; в группу могут добавляться новые РТС и пр. Таким образом, в ПО могут потребоваться специальные блоки обеспечения динамического реконфигурирования состава групп РТС по ходу выполнения поставленных задач, а также учета изменения возможностей отдельных РТС при получении ими повреждений, уменьшении ресурсов энергопитания.

Еще более сложной задачей может быть разработка ПО не только для нескольких РТС, действующих в составе группы, но и нескольких взаимодействующих друг с другом групп РТС, т.е. с выделением трех иерархических уровней устройств.

10. Отладка создаваемого ПО для РТС обычно последовательная (по мере написания кода). При этом для тех блоков ПО, которые пока не разработаны, может быть целесообразным временное использование блоков-имитаторов, которые впоследствии будут заменены фрагментами реального кода.

Иногда итоговая отладка является отдельным этапом создания ПО, который должен выполняться после разработки кода. Это касается, в частности, стыковочной отладки крупных блоков ПО, разработанных для автономных РТС; стыковочной отладки ПО, предназначенного для функционально разных РТС, действующих в составе групп устройств и др.

Предварительным этапом отладки ПО является устранение всех синтаксических ошибок в кодах программ, которые обнаруживает компилятор. При этом целесообразно сразу проверять весь код на наличие аналогичных ошибок. Затем в рамках отладки осуществляется выявление и устранение логических ошибок, в т.ч.

путем задания различных входных данных; вывода промежуточных результатов (отладочных сообщений) с использованием операторов, временно помещенных в состав кода. Эти сообщения могут выдаваться и на дисплей, входящий в состав РТС или дополнительно подключаемый к ней только на период отладки. Более простой вариант – выдача кодовых совокупностей звуковых сигналов, в т.ч. с разной продолжительностью, тональностью и пр. Также может быть предусмотрен вывод отладочных сообщений через интерфейс связи (USART, USB), не используемый при штатной эксплуатации РТС; в дополнительные окна ИСР и пр. При отладке может быть предусмотрен и прием ПО ответов от лица, осуществляющего отладку в диалоговом режиме. Это иногда дает дополнительные возможности по отслеживанию логики программы по сравнению с изменением состава входных данных и включением в код отладочных операторов.

После завершения отладки соответствующих блоков ПО (или ПО в целом) отладочные операторы могут быть из кода удалены. Однако если они не оказывают серьезного влияния на производительность РТС и не снижают их ИБ, то такие операторы могут быть сохранены в финальной версии ПО со следующими целями: диагностикой неисправностей РТС, в т.ч. при самотестировании устройств по запросам людей-операторов; поддержкой возможной в будущем модернизации ПО, в т.ч. в отношении его отладки; проверкой соблюдения условий ИБ при возникновении новых типов угроз для ПО РТС.

Включение или отключение отладочных-операторов, находящихся в составе ПО, может управляться за счет ввода в составе исходных данных специального кода. Промежуточным вариантом между удалением и сохранением отладочных операторов в ПО может быть их комментирование (т.е. перевод из категории исполняемых в текстовые комментарии).

Иногда для отладки ПО РТС используются специальные программно-аппаратные комплексы [30]. В их составе может использоваться дополнительное ПО, разработанное специально для их использования.

Особую сложность представляет отладка блоков ПО, использующих операторы взаимодействия с внешними устройствами, другими РТС и / или ЦУ. Здесь возможно два подхода: 1) применение реальных устройств и визуальный / аудиовизуальный контроль их работы; 2) программная эмуляция работы таких устройств, включая нажатия кнопок лицами, осуществляющими управление РТС; оценки напряжения питания на аккумуляторе, обеспечивающем энергопитание РТС; поступление информации с СС и др. Реализация второго варианта возможна, например, в программах типа “Proteus”. Однако эмуляция изображений, поступающих с СТЗ РТС [14, 25] для проверки эффективности их обработки в разрабатываемом ПО, может представлять существенные сложности. Отметим также ряд разработок, указанных в [24].

По крайней мере, в случае А при отладке может быть целесообразной временная вставка в код ПО операторов задержек по времени, с тем чтобы можно было успеть проследить и осмыслить реакции внешних устройств (и / или других РТС) на выполнение операторов (блоков операторов) в отлаживаемой программе.

В общем случае при автономной отладке ПО для РТС должны быть проверены все логические ветви, в т.ч. соответствующие различным вариантам входной информации, движения механических частей РТС, появления перед ними препятствий, поступления различных изображений с СТЗ и пр. Однако если таких логических ветвей слишком много, а время, имеющееся в распоряжении разработчиков, ограничено, то нередко выполняется лишь частичная проверка типичных штатных вариантов работы РТС; предельных значений для основных параметров; основных нештатных ситуаций, которые можно предусмотреть. Это, естественно, снижает надежность разработанного ПО и может приводить к выявлению ошибок и недочетов уже при комплексных приемо-сдаточных испытаниях РТС или даже на стадии их эксплуатации.

Отладка ПО для РТС отличается от отладки других видов программ также тем, что при каждом изменении текста программы обычно необходима не только ее перекомпиляция, но и запись в память МП (или МК). Это требует дополнительного времени, иногда применения специальных программаторов. Аналоговые элементы электрических схем РТС обычно вообще невозможно отлаживать в привычном для разработчиков ПО режиме, а их симуляция (эмуляция) [24] может носить лишь приблизительный характер. Эти факторы, конечно, удлиняют продолжительность отладки ПО для РТС, могут снижать надежность разработок, иногда их уровни ИБ.

11. Подэтап оценки фактической вычислительной эффективности алгоритмов необходим лишь для некоторых видов ПО для РТС, прежде всего исследовательского и производственного назначения. Для оценок продолжительностей исполнения в ПО могут временно включаться дополнительные операторы, а результаты оценок выдаваться, например, на штатный дисплей для РТС или специально подключаемый при тестировании. Для оценки эффективности разработанного ПО могут использоваться и специальные математические методы, включая сети Петри – Маркова [22].

Испытания ПО в составе РТС. По месту проведения испытаний они могут быть категоризованы следующим образом: в «полевых условиях», в т.ч. на специальных полигонах, в лабораториях. В последнем случае может выполняться физическая имитация условий среды эксплуатации РТС и внешних воздействий на них. Это позволяет проверить различные ситуации, в т.ч. и редко встречающиеся в полевых условиях.

Дополнительным направлением испытаний ПО может быть использование имитационных компьютерных моделей (ИКМ) для РТС, работающих под управлением разработанного ПО. При этом ИКМ

могут выполнять «подыгрывающие функции»; моделировать как условия внешней среды, так и результаты взаимодействия с ней РТС, работающей под управлением созданного ПО. Имитационное моделирование позволяет исследовать поведение РТС в опасных условиях, в случаях частичного разрушения (повреждения) РТС и пр.

Категорирование испытаний по видам выполним так.

А. Промежуточные испытания РТС (по ходу выполнения разработки ПО или РТС в целом) могут представлять собой часть процесса создания ПО; относиться к его отдельным блокам и / или использованию БД; относиться к ПО в целом. Целью таких испытаний может быть отладка процессов перемещений частей РТС, управляемых ПО; проверка отсутствия утечек информации с РТС по ЭМК; оценка адаптивности обучаемого ПО для РТС к внешним воздействиям и пр.

Б. Итоговые испытания – по завершении разработки РТС могут осуществляться по следующим направлениям: проверка ПО в составе РТС в отношении его функциональности, скорости работы, совместимости с механическими частями конструкций; оценка корректности и точности управления ходовыми частями мобильных РТС, его манипуляторами и пр.; проверка правильности и эффективности процессов адаптации ПО РТС к внешним условиям, в т.ч. путем корректировки БД, обучения нейронных сетей и пр.; оценка устойчивости разработанного ПО к злонамеренным внешним воздействиям, в т.ч. угрозам заражения вирусами, к перехвату управления РТС; оценка защищенности БКС от утечек информации и пр.

Содержание итоговых испытаний отличается для разных типов РТС.

Б1. Испытания ПО для РТС, предназначенных для серийного выпуска. Цель – подтверждение запланированной функциональности разработанного ПО; его совместимости с конструкциями РТС, их СС и / или СТЗ, электромеханическими частями; проверка соблюдения требований ИБ в отношении исключения утечек информации по ЭМК, устойчивости к внешним электромагнитным воздействиям, попыткам несанкционированного управления РТС и / или блокирования управления со стороны ЦУ.

Б2. Испытания ПО в составе уникальных РТС, предназначенных для использования конкретным заказчиком. Программы проведения таких испытаний должны быть заранее разработаны и согласованы с заказчиком, а его представители могут и должны участвовать в проведении приемо-сдаточных испытаний. При особо сложных РТС испытания их подсистем (в т.ч. связанные с использованием ПО) могут производиться по отдельности. Могут использоваться методы испытаний СТС [10, 31, 32].

Б3. Проверка работоспособности и функциональности ПО для РТС учебного назначения, в т.ч. в связи с защитой бакалаврских работ, магистерских диссертаций. В этом случае обучающиеся обычно сами разрабатывают сценарии демонстрационных проверок работы РТС. Однако при защитах могут выполняться и дополнительные проверки по требованиям комиссий, осуществляющих государственную аттестацию.

Б4. Проверка (испытания) ПО в составе РТС, предназначенных для участия в различных видах соревнований по робототехнике [23] как отдельных единиц РТС, так и их групп (например, команд роботов для участия в футбольных соревнованиях). Такая проверка осуществляется обычно самими соревнующимися.

В. Проверка серийно выпускаемых РТС по окончании их изготовления, в т.ч. проверка ПО. Такую проверку целесообразно проводить по заранее утвержденной, унифицированной программе испытаний. Основная цель – подтверждение того, что серийная продукция по своим характеристикам не отличается от ранее испытанного образца, утвержденного для выпуска. Для обеспечения качества выпускаемых РТС могут использоваться меры управления качеством: введение или ужесточение пооперационного контроля производимой продукции; переход от выборочного контроля блоков РТС к сплошному; усиление контроля комплектующих и пр. При этом предполагается, что ПО для РТС остается неизменным, а проверка его аутентичности разработкам, выполненным ранее, может осуществляться не только по размерам файлов, но и по их CRC.

Поскольку изготовленные РТС обычно являются СТС, целесообразно проведение испытаний всех единиц продукции. Однако часть РТС могут испытываться в минимальном объеме, а часть – в более полном. При выявлении несоответствий изготовленных изделий установленным требованиям дополнительные испытания могут осуществляться для всех единиц РТС в партии.

Подчеркнем, что с течением времени может меняться среда эксплуатации РТС, а также могут появляться новые виды угроз для РТС, ЦУ и каналов связи, которые не были предусмотрены в ранее разработанном ПО. Поэтому если программы испытаний фактически меняются, то результаты могут стать неудовлетворительными. Как следствие может потребоваться модернизация ПО или РТС в целом.

При достаточно больших объемах выпуска РТС или в случае уникальных изделий высокой стоимости целесообразна разработка, изготовление и использование специальных проверочных стендов, в составе которых может применяться специальное ПО, в т.ч. и непосредственно взаимодействующее с ПО на проверяемых РТС.

Г. Испытания ПО в составе РТС после выполнения модернизации ПО и / или замены СС, навигационных систем, СТЗ, исполнительных устройств; изменения номенклатуры РТС, входящих в состав их групп и пр. Типичные причины необходимости модернизации ПО РТС в процессе их эксплуатации рассмотрены в следующем разделе. Возможны следующие основные варианты модернизации ПО: ча-

стичная замена фрагментов кода программы без изменения структуры ПО в целом; глубокая модернизация ПО, при которой изменяется ее структура, появляются дополнительные функции, изменяются существующие функции и пр.; изменение состава библиотек, используемых в ПО; полная замена ПО на новое, в котором используются лишь отдельные блоки из предыдущей версии ПО.

Глубина модернизации ПО значительно влияет на объемы проведения его испытаний после корректировки. Обычно чем больше объемы изменений в ПО, тем больше должны быть и объемы его испытаний. Однако нередко даже при минимальных изменениях в ПО возможны побочные эффекты, которые не будут своевременно выявлены при проведении небольших объемов испытаний.

Особо тщательно необходимо проводить испытания после модификации ПО для уникальных РТС, а также устройств, предназначенных для крупносерийного выпуска и / или применения заранее не определенным кругом пользователей. Подчеркнем, что здесь речь идет об испытаниях не только собственно ПО, но и РТС в целом.

Д. Проверка ПО на эксплуатируемых РТС:

- Периодическая проверка – если она предусматривается регламентом эксплуатации РТС в целом или только в отношении установленного на них ПО. При этом целесообразно использовать унифицированные наборы контрольных операций, предпочтительно выполняемых в автоматическом или автоматизированном режиме.

- Проверки работоспособности единиц РТС после воздействия на них нештатных ситуаций, которые могли повредить их конструкции; изменить ПО или БД, заложенные в память РТС и пр.

- Проверки ПО, связанные с изменением условий эксплуатации РТС, в т.ч. при появлении новых видов угроз ИБ, которые не существовали ранее и пр.

- Проверки работоспособности ПО после замены деталей (блоков) на иные, чем были первоначально предусмотрены в конструкции, например, в порядке ремонта.

Отметим, что для указанных выше видов проверок ПО в местах производства РТС могут использоваться те же проверочные стенды, что и для проведения испытаний после выпуска РТС.

Рассмотрим теперь кратко задачу распределения затрат (усилий) создателей ПО между такими этапами: его разработкой; испытаний ПО после окончания его разработки, устранением недочетов в ПО на этапе его эксплуатации. Для этой цели используем математическую модель, учитывающую как затраты, так и возможные ущербы. Будем считать, что затраты, связанные с ПО, имеют такие компоненты: на создание ПО (Z_{soz}^+); на проведение испытаний ПО (Z_{isp}^+). Примем ограничения для обоих типов затрат в виде интервалов

$$Z_{soz}^{(min)+} \leq Z_{soz}^+ \leq Z_{soz}^{(min)+}; Z_{isp}^{(min)+} \leq Z_{isp}^+ \leq Z_{soz}^{(min)+}. \quad (1)$$

Пусть Ψ есть сумма затрат на разработку и испытания ПО и вероятных ущербов в процессе его эксплуатации, связанных с некачественно разработанным ПО и неполнотой проведения его испытаний. При этом для этапа эксплуатации мы будем учитывать не только прямые ущербы, связанные с выявлением в разработанном ПО недочетов, но и затраты на их устранение. Примем, что в диапазонах изменения затрат, учитываемых в (1), значения Ψ описываются формулой

$$\Psi = NV \left(U_{exp}^* - f_1(Z_{soz}^+ - Z_{soz}^{(min)+}) - f_2(Z_{isp}^+ - Z_{isp}^{(min)+}) - f_3((Z_{soz}^+ - Z_{soz}^{(min)+}) \times (Z_{isp}^+ - Z_{isp}^{(min)+})) \right), \quad (2)$$

где N – количество предполагаемых к выпуску (эксплуатации) РТС; V – предполагаемая продолжительность эксплуатации каждой единицы РТС; U_{exp}^* – величина ущерба, за единицу времени по одному РТС, соответствующая минимальным величинам затрат, соответственно, на разработку и испытания ПО; f_1, f_2 – в общем случае нелинейные функции, описывающие уменьшение величины ущерба (за единицу времени в расчете на одну РТС) в зависимости от объемов затрат, соответственно, на разработку ПО и проведение его итоговых испытаний после окончания разработки; f_3 – функция, описывающая синергетические эффекты от этих двух видов затрат (также в расчете на одну РТС за единицу времени). Виды этих трех функций и значения коэффициентов в них могут быть оценены экспертно и/или на основе опыта разработки/испытаний/эксплуатации аналогичного ПО для РТС. В простейшем случае для U_{exp}^* можно принять аддитивную модель

$$U_{exp}^* = \sum_{i=1}^I (P_i U_i), \quad (3)$$

где I – общее количество видов неблагоприятных событий (НС); P_i – вероятность реализации i -ого НС для одного РТС за единицу времени; U_i – ущерб при однократной реализации i -го НС (эти величины также могут быть оценены экспертно). Для простоты мы также считаем, что однократные реализации НС не приводят к изменениям вероятностей реализации НС в последующий период и ущербов от их реализации, т.е. принимаем модель ущербов без последствия. При этом мы считаем, что дополнительные

затраты (сверх минимальных, включенных в формулу (1)) могут приводить как к уменьшению вероятностей НС, так и ущербов от них.

Оптимизация решений по разработке / испытаниям / эксплуатации ПО для РТС сводится к минимизации величины Ψ при фиксированных значениях N, V . Такая оптимизация может выполняться как на дискретных множествах их значений, так и непрерывных. При этом могут использоваться различные методы условной оптимизации, включая градиентные. Однако необходимо понимать, что описанная задача оптимизации, как правило, осуществляется в нечетких условиях.

Результаты решения оптимизационной задачи могут использоваться при планировании распределения финансовых и трудовых затрат по этапам ЖЦ ПО; оценки количества персонала, необходимого для разработки ПО для РТС; определении необходимых календарных сроков выполнения разработки ПО и ее испытаний; обосновании длительности ЖЦ ПО для РТС и т.д.

Варианты документирования результатов испытаний ПО для РТС:

1. Составление актов (протоколов) испытаний ПО и непосредственное включение в эти акты всего необходимого текстового, числового, иллюстративного материала.

2. Фиксация (фото съемка, видеозапись) поведения РТС, управляемого разработанным ПО. Возможно, приложение этих фото, видеозаписей к протоколам испытаний.

3. Запись телеметрической информации, поступающей от РТС в процессе проведения испытаний. Обычно это требует создания специальных блоков ПО, обеспечивающих передачу телеметрии без существенного увеличения продолжительности реализации основных действий, выполняемых ПО.

Во всех трех случаях необходима привязка результатов испытаний к астрономическому времени, местам проведения испытаний.

Этап эксплуатации программного обеспечения для РТС. Здесь также возможно несколько вариантов:

1. ПО остается неизменным в течение всего срока эксплуатации РТС.

2. Исполняемая программа не меняется, но меняется БД, используемая в составе ПО, например, по результатам эксплуатации единицы РТС. На основе этой БД ПО самостоятельно принимает и реализует решения, заложенные в алгоритм.

3. Совместно с ПО используется корректируемая база знаний, в т.ч. построенная с использованием нейронных сетей с забыванием информации.

4. Копии разработанного ПО эксплуатируют несколько РТС, взаимодействующих друг с другом по информации и управлению. При этом количество РТС в таких группах в общем случае может меняться. Для таких групп РТС могут использоваться алгоритмы автоматической выработки коллективных решений, в т.ч. и на основе индивидуальных БД, сформированных для каждой из РТС.

Как и все технические системы РТС имеют ограниченный ЖЦ [4]. Это обуславливается не только физическим старением устройств, но и моральным, в т.ч. и моральным старением ПО. Управление длительностью ЖЦ РТС возможно различными способами: своевременное сервисное обслуживание; переуплотнение использования РТС; доработка их механических частей; модернизацию ПО и др. Для управления ЖЦ ПО существуют специальные методики и ПО (“Telelogic Harmony”, среды “IBM Rhapsody”, “QNX Momentics IDE”) [4].

Основные причины, которые могут обусловить необходимость модернизации ПО на этапе его эксплуатации: 1) появление новых задач для РТС, которые не были предусмотрены при первоначальной разработке; 2) выявление функциональных недочетов в существующем ПО, в т.ч. некорректных или несвоевременных реакций РТС на изменения окружающих условий; 3) изменение операционных систем и ПО, используемых в ЦУ РТС; 4) Изменения ПО на внешних устройствах, с которыми взаимодействуют РТС; 5) необходимость корректировки ПО при переносе его на новые аппаратные средства, расширении номенклатуры датчиков СС, исполнительных устройств и пр.; 6) необходимость создания дополнительных интерфейсных блоков для обеспечения взаимодействия РТС с новыми информационными системами организаций, ЦУ и пр.; 7) появление необходимости перенастроек блоков ПО, обеспечивающих для РТС избегание потенциально опасных для них ситуаций, при необходимости, для действий РТС в таких ситуациях; 8) Появление новых угроз ИБ для самих РТС и / или каналов приема-передачи данных, в т.ч. от РТС к ЦУ и от ЦУ к РТС, а также между группами взаимодействующих между собой РТС; 9) в рамках принятия решений о продлении сроков эксплуатации РТС за счет модернизации ПО, в т.ч. с учетом рисков таких решений [28].

Для серийно выпускаемых РТС модернизация ПО может осуществляться в централизованном порядке, а обеспечение этим ПО пользователей РТС может выполняться в порядке гарантийного обслуживания или за дополнительную плату после окончания срока гарантии. Для РТС учебного назначения вопросы модернизации ПО менее актуальны, чем для других типов устройств.

Отметим актуальность для этапа эксплуатации ПО таких вопросов. А. Необходимость обеспечения информационной поддержки пользователей (эксплуатантов) РТС со стороны разработчиков в отно-

шении использования созданного ими ПО. Б. Исключение возможностей несанкционированного доступа к кодам ПО для эксплуатируемых РТС, в т.ч. за счет того, что в составе РТС распространяются только ПО в виде исполнимых модулей, но не исходных кодов. В. Защита РТС от несанкционированного изменения эксплуатируемого на них ПО, в т.ч. за счет аппаратных решений, проведения исполнимыми программами самотестирования при запуске (см. выше).

Прекращение эксплуатации программного обеспечения для РТС. Возможные причины прекращения эксплуатации ПО для РТС: необходимость замены ПО; прекращение эксплуатации РТС в целом.

В типичных случаях модернизация или замена ПО для РТС осуществляются по мере необходимости: заранее планировать эти процессы может быть сложно. Выбор между решением по модернизации ПО и его замене обычно осуществляется в условиях неопределенности, т.к. обычно заранее неизвестно, сколько времени может потребоваться до следующей модернизации (или замены ПО), а также того, сколько еще может прослужить РТС с учетом физического и морального износа. Отметим, что в случае серийно изготавливаемых РТС выпуск для них запчастей осуществляется лишь ограниченное время после прекращения выпуска устройств. Таким образом, рациональность затрат на модернизацию или замену ПО должна увязываться с предполагаемыми сроками последующей эксплуатации РТС; интенсивностями их использования; затратами на сервисное обслуживание и ремонт механических частей, электроприводов и пр.

В случае временной консервации единиц РТС (т.е. снятия их с эксплуатации) меры в отношении ПО могут включать следующее: сохранение контрольных копий ПО, в т.ч. БД, накопленных в процессе эксплуатации; обеспечение безопасных условий хранения РТС (или только электронных блоков, являющихся носителями ПО), при которых посторонние лица не будут иметь к ним доступа.

При проведении утилизации РТС без уничтожения МП (например, его предполагается использовать в составе другой разработки) может быть целесообразным специально стирать ПО из памяти МП (или МК) путем многократной записи случайных кодов. Тем самым будет исключен канал утечки информации в отношении того ПО, которое не предназначено для открытого распространения, повторного использования.

Выводы. 1. Предложена классификация стадий (этапов) жизненного цикла ПО для РТС, учитывающая специфику разработки, испытаний и использования такого ПО. 2. Для каждого из выделенных этапов ЖЦ ПО для РТС рассмотрена структура (состав) подэтапов, возможные виды рисков для этих подэтапов. 3. Проанализированы потенциально возможные и практически целесообразные меры риск-менеджмента в отношении разработки, испытаний и эксплуатации ПО для РТС. 4. Рассмотрена постановка задачи «оптимизации распределения затрат» между этапами разработки ПО, его испытаний, устранения недочетов в ходе эксплуатации. 5. Показано, что целесообразность модернизации или прекращения эксплуатации ПО для РТС может определяться не только его функциональностью и качеством, но и рядом «внешних» по отношению к ПО причин (факторов), включая физическое и моральное старение РТС в целом.

Список литературы

1. Антонов А. В. Система управления трехопорным колесно-шагающим роботом / А. В. Антонов, С. А. Воротников, Н. А. Выборнов // Прикаспийский журнал: управление и высокие технологии. – 2016. – № 2. – С. 58–69 ([http://hi-tech.asu.edu.ru/files/2\(34\)/58-69.pdf](http://hi-tech.asu.edu.ru/files/2(34)/58-69.pdf)).
2. Айтбекова Д. С. Интеграция и тестирование модифицированных процедур с общей программой микроконтроллера, осуществляющего основные функции управления сервоприводами мобильного робота / Д. С. Айтбекова, Н. А. Гумело, А. С. Дервоедов, А. А. Карпеева // Научный альманах. – 2016. – № 8–1 (22). – С. 168–171.
3. Балыбердин В. А. Анализ основных процессов обеспечения надежности программных средств АСУ специального назначения / В. А. Балыбердин, А. М. Белевцев, О. А. Степанов // Известия ЮФУ. Технические науки. – 2015. – № 3 (164). – С. 62–70.
4. Большаков А. А. Программный комплекс управления жизненным циклом мехатронных систем / А. А. Большаков, Д. Ю. Петров // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2012. – № 2. – С. 20–26.
5. Брискин Е. С. Построение алгоритмов движения шагающего робота / Е. С. Брискин, Я. В. Калинин, А. В. Малолетов, Н. Г. Шаронов // Экстремальная робототехника. – 2014. – Т. 1. № 1. – С. 293–297.
6. Брумштейн Ю. М. Робототехнические системы: вопросы разработки / Ю. М. Брумштейн, М. А. Ильменский, И. В. Колесников // Интеллектуальная собственность. Авторское право и смежные права. – 2016. – № 9. – С. 49–64.
7. Брумштейн Ю. М. Робототехнические системы: вопросы использования / Ю. М. Брумштейн, М. А. Ильменский, И. В. Колесников // Интеллектуальная собственность. Авторское право и смежные права. – 2016. – № 9. – С. 49–64.
8. Брумштейн Ю. М. Программы для ЭВМ и связанные с ними объекты. Анализ понимания терминов в законодательстве и сфере информационных технологий / Ю. М. Брумштейн // Интеллектуальная собственность. Авторское право. – 2008. – № 10. – С. 26–38 ([http://hi-tech.asu.edu.ru/files/4\(20\)/35-45.pdf](http://hi-tech.asu.edu.ru/files/4(20)/35-45.pdf)).
9. Брумштейн Ю. М. Одно- и многомерные временные ряды: анализ возможных методов оптимизации отсчетов и оценки характеристик / Ю. М. Брумштейн, М. В. Иванова // Прикаспийский журнал: управление и высокие технологии. – 2012. – № 4. – С. 35–44.
10. Бурба А. Оценка эффективности испытаний сложных технических систем / А. Бурба // Стандарты и качество. – 2013. – № 5 (911). – С. 58–59.
11. Венславский В. Б. Коммуникационные проблемы подготовки студентов к обучению школьников электронике и робототехнике / В. Б. Венславский, И. В. Ладыгина // Инновационные технологии в технике и образовании :

материалы VI Международной научно-практической конференции. – Забайкальский государственный университет, 2015. – С. 56–61.

12. Гомилко С. И. Разработка программного обеспечения команды роботов-футболистов лиги HUMANOID KIDSIZE чемпионата ROBOCUP / С. И. Гомилко, Д. В. Жулаева, Д. И. Ример, Е. С. Шандаров, Д. О. Якушин // Электронные средства и системы управления. – 2015. – № 1–2. – С. 104–108.

13. Ильяшенко А. С. Система управления распределенным коллективом роботов / А. С. Ильяшенко, Л. Ю. Лабошин, А. А. Лукашин // Экстремальная робототехника. – 2015. – № 1 (1). – С. 320–323.

14. Исупова Т. Д. Разработка подсистемы технического зрения робота андроидного типа как объекта двойного назначения / Т. Д. Исупова // Известия Тульского государственного университета. Технические науки. – 2015. – № 11–2. – С. 209–215.

15. Кабак И. С. Математическая модель для прогнозирования надежности программного обеспечения / И. С. Кабак // Вестник МГТУ. – 2014. – № 1 (28). – С. 123–126.

16. Кайснер Э. Робототехника: прорывные технологии, инновации, интеллектуальная собственность / Э. Кайснер, Д. Раффо, С. Вунш-Винсент // Форсайт. – 2016. – Т. 10, № 2. – С. 7–27.

17. Климов М. А. Способ калибровки систем локальной навигации мобильных роботов / М. А. Климов, С. А. Воротников, Н. А. Выборнов // Прикаспийский журнал: управление и высокие технологии. – 2017. – № 1. – С. 106–115.

18. Кожемякин И. В. Разработка подводного робототехнического комплекса, с использованием открытых средств моделирования движения, дополненных моделью гидроакустического взаимодействия / И. В. Кожемякин, И. А. Путинцев, Н. Н. Семенов, М. Н. Чемоданов // Известия ЮФУ. Технические науки. – 2016. – № 1 (174). – С. 88–101.

19. Колкер А. Б. Исследование вариантов создания интеллектуальных систем робототехники на базе одно-платных компьютеров и свободных операционных систем / А. Б. Колкер, Д. А. Ливенец, А. И. Кошелева, В. А. Жмудь // Автоматика и программная инженерия. – 2012. – № 1 (1). – С. 84–98.

20. Коньшев Д. В. Управление мимическим аппаратом сервисных роботов при синтезе эмоций / Д. В. Коньшев, С. А. Воротников, Н. А. Выборнов // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 3. – С. 216–229 ([http://hi-tech.asu.edu.ru/files/3\(27\)/216-229.pdf](http://hi-tech.asu.edu.ru/files/3(27)/216-229.pdf)).

21. Костюкова А. В. Основные элементы блок-схемы программного обеспечения робота-горноспасателя / А. В. Костюкова, Е. В. Пикина, К. Н. Чибинев, М. Е. Шматько // Горный информационно-аналитический бюллетень. – 2016. – № 3. – С. 384–388.

22. Ларкин Е. В. Оценка эффективности программного обеспечения робота с использованием сетей Петри-Маркова / Е. В. Ларкин, В. В. Котов, Н. А. Котова // Известия Тульского государственного университета. Технические науки. – 2013. – № 9–2. – С. 156–162.

23. Медведева С. И. Робототехнические соревнования как механизм профориентации подростков и обучения специалистов / С. И. Медведева, С. О. Шепелёв, А. А. Некрасов // Инновационные подходы к решению технико-экономических проблем. – 2015. – С. 227–229.

24. Михайлова У. В. Программные решения для разработки архитектуры системы управления роботом / У. В. Михайлова, Е. А. Михайлов, А. С. Сарваров // Электротехнические системы и комплексы. – 2013. – № 21. – С. 111–117.

25. Нгуен Туан Зунг. Распознавание объектов в системе технического зрения мобильного робота : использование библиотеки FLANN и алгоритма SURF / Нгуен Туан Зунг, И. А. Щербатов // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 4. – С. 65–76 ([http://hi-tech.asu.edu.ru/files/4\(28\)/65-76.pdf](http://hi-tech.asu.edu.ru/files/4(28)/65-76.pdf)).

26. Отрадных К. К. Построение интеллектуального программного обеспечения для управления двуногим шагающим роботом с поступательными кинематическими парами в суставах ног / К. К. Отрадных // Известия высших учебных заведений. Машиностроение. – 2012. – № 11. – С. 42–45.

27. Петрина А. М. О взаимодействии интеллектуальных роботов, программного обеспечения и компьютеров / А. М. Петрина // Научно-техническая информация. Серия 2: Информационные процессы и системы. – 2015. – № 2. – С. 31–38.

28. Полянский В. И. Проблемные вопросы управления рисками при разработке, испытаниях и продлении ресурса сложных технических систем военного назначения / В. И. Полянский // Проблемы информатики. – 2012. – № 4 (16). – С. 89–92.

29. Рогаткин Д. А. Медико-технические требования к автономным мобильным сервисным медицинским роботам / Д. А. Рогаткин, Л. Г. Лапаева // Робототехника и техническая кибернетика. – 2016. – № 2 (11). – С. 45–51.

30. Серов А. Н. Программно-аппаратный комплекс контроля и отладки программного обеспечения вычислительного устройства для наземного мобильного робота / А. Н. Серов, Ю. В. Савченко, А. В. Шипатов, А. В. Сотников // Известия высших учебных заведений. Электроника. – 2014. – № 6 (110). – С. 43–50.

31. Старусев А. В. Метод оценки и обеспечения качества испытаний автоматизированных систем / А. В. Старусев // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 4 (28). – С. 197–204 ([http://hi-tech.asu.edu.ru/files/4\(28\)/197-204.pdf](http://hi-tech.asu.edu.ru/files/4(28)/197-204.pdf)).

32. Старусев А. В. планирования испытаний сложных технических систем / А. В. Старусев, В. И. Лобейко, С. П. Литвинов // Известия Волгоградского государственного технического университета. – 2015. – № 2 (157). – С. 90–93.

33. Шматько М. Е. Основные требования к программному обеспечению робота-горноспасателя / М. Е. Шматько, Е. В. Пикина, Н. Н. Чибинёв, К. Н. Чибинёв // Горный информационно-аналитический бюллетень. – 2016. – № 1. – С. 135–138.

34. Учаев Д. Ю. Анализ и управление рисками, связанными с информационным обеспечением человеко-машинных АСУ технологическими процессами в реальном времени / Д. Ю. Учаев, Ю. М. Брумштейн, И. М. Ажмухадеев, О. М. Князева, И. А. Дюдиков // Прикаспийский журнал: управление и высокие технологии. – 2016. – № 2 (34). – С. 82–97.

35. Черников Б. В. Управление качеством программного обеспечения / Б. В. Черников. – Москва : ФОРУМ :

ИФРА-М, 2012. – 240 с.

36. Betty H. C. Cheng and 28 more. Software Engineering for Self-Adaptive Systems: A Research Roadmap / H. C. Betty // Lecture Notes in Computer Science. – 2009. – Vol. 5525. – P. 1–26.

37. Medvidovic Nenand. Engineering Heterogeneous Robotics Systems: A Software Architecture-Based Approach / Medvidovic Nenand and others // Computer. – May 2011. – Vol. 44, issue 5. – P. 62–71.

38. Schlegel C. (). Model-driven software systems engineering in robotics: Covering the complete life-cycle of a robot. Special Issue: Robot Control Architectures / C. Schlegel, A. Lotz, M. Lutz, K. Berns, R. Dillmann, E. Maehle // Information Technology. – 2015. – Vol. 57(2). – P. 85–98. DOI: 10.1515/itit-2014-1069.

39. Zimin G. A. Vizual dataflow language for educational robots programming / G. A. Zimin, D. A. Mordvinov // Труды Института системного программирования РАН. – 2016. – Vol. 28, № 2. – P. 45–62.

40. Yim Mark. Modular Self-Reconfigurable Robot Systems (Grand Challenges of Robotics) / Yim Mark and others // IEEE Robotics & Automation Magazine. – March 2007, vol. 14, issue 1. – P. 43–52.

References

1. Antonov A. V., Vorotnikov S. A., Vybornov N. A. Sistema upravleniya trekhopornym kolesno-shagayushchim robotom [The control system of three-point wheel-legged robot]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control And High Technologies], 2016, no. 2, pp. 58–69 ([http://hi-tech.asu.edu.ru/files/2\(34\)/58-69.pdf](http://hi-tech.asu.edu.ru/files/2(34)/58-69.pdf)).

2. Aytbekova D. S., Gumelo N. A., Dervoedov A. S., Karpeeva A. A. Integratsiya i testirovanie modifitsirovannykh protsedur s obshchey programмой mikrokontrollera, osushchestvlyayushchego osnovnye funktsii upravleniya servoprivodami mobilnogo robota [Integration and testing of the modified procedures into the general program of the microcontroller which is realizing basic functions of control of servo actuators of the mobile robot]. *Nauchnyy almanakh* [Scientific Almanac], 2016, no. 8–1 (22), pp. 168–171.

3. Balyberdin V. A., Belevtsev A. M., Stepanov O. A. Analiz osnovnykh protsessov obespecheniya nadezhnosti programnykh sredstv ASU spetsialnogo naznacheniya [Analysis of the main processes of support of reliability of software of ACS of a special purpose]. *Izvestiya YuFU. Tekhnicheskije nauki* [Proceedings of South Federal University. Engineering], 2015, no. 3 (164), pp. 62–70.

4. Bolshakov A. A., Petrov D. Yu. Programmnyy kompleks upravleniya zhiznennym tsiklom mekhatronnykh sistem [Program complex of control of life cycle mechatronic systems]. *Vestnik Astrakhanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of the Astrakhan State Technical University. Series: Control, Computing Equipment and Informatics], 2012, no. 2, pp. 20–26.

5. Briskin Ye. S., Kalinin Ya. V., Maloletov A. V., Sharonov N. G. Postroenie algoritmov dvizheniya shagayushchego robota [Creation of algorithms of the legged robot movement]. *Ekstremalnaya robototekhnika* [Extreme Robotics], 2014, vol. 1, no. 1, pp. 293–297.

6. Brumshteyn Yu. M., Ilmenskiy M. A., Kolesnikov I. V. Robototekhnicheskije sistemy: voprosy razrabotki [Robotic systems: questions of development]. *Intellektual'naya sobstvennost. Avtorskoe pravo i smezhnye prava* [Intellectual Property. Copyright and Adjacent Rights], 2016, no. 9, pp. 49–64.

7. Brumshteyn Yu. M., Ilmenskiy M. A., Kolesnikov I. V. Robototekhnicheskije sistemy: voprosy ispolzovaniya [Robotic systems: questions of usage]. *Intellektual'naya sobstvennost. Avtorskoe pravo i smezhnye prava* [Intellectual Property. Copyright and Adjacent Rights], 2016, no. 9, pp. 49–64.

8. Brumshteyn Yu. M. Programmy dlya EVM i svyazannye s nimi obekty. Analiz ponimaniya terminov v zakonodatel'stve i sfere informatsionnykh tekhnologii [Computer programs and related objects. The analysis of understanding of terms in the legislation and the sphere of information technologies]. *Intellektual'naya sobstvennost. Avtorskoe pravo i smezhnye prava* [Intellectual Property. Copyright and Adjacent Rights], 2008, no. 10, pp. 26–38.

9. Brumshteyn Yu. M., Ivanova M. V. Odno- i mnogomernye vremennye ryady: analiz vozmozhnykh metodov optimizatsii otschetov i otsenki kharakteristik [One- and multivariate time series: analysis of possible methods for optimization of the counting and characteristics evaluation]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control And High Technologies], 2012, no. 4, pp. 35–44 ([http://hi-tech.asu.edu.ru/files/4\(20\)/35-45.pdf](http://hi-tech.asu.edu.ru/files/4(20)/35-45.pdf)).

10. Burba A. Otsenka effektivnosti ispytaniy slozhnykh tekhnicheskikh sistem [Evaluation of difficult technical systems tests efficiency]. *Standarty i kachestvo* [Standards and Quality], 2013, no. 5 (911), pp. 58–59.

11. Venslavskiy V. B., Ladygina I. V. Kommunikatsionnye problemy podgotovki studentov k obucheniyu shkolnikov elektronike i robototekhnike [Communication problems of training of students for training of school students in electronics and robotic technology]. *Innovatsionnye tekhnologii v tekhnike i obrazovanii : materialy VI Mezhdunarodnoy nauchno-prakticheskoy konferentsii* [Innovative Technologies in Technique and Education. Proceedings of the VI International Scientific and Practical Conference], 2015, pp. 56–61.

12. Gomilko S. I., Zhulaeva D. V., Rimer D. I., Shandarov Ye. S., Yakushin D. O. Razrabotka programmogo obespecheniya komandy robotov-futbolistov ligi HUMANOID KIDSIZE chempionata ROBOCUP [Software development of a command of robots football players of league HUMANOID KIDSIZE of the ROBOCUP championship]. *Elektronnyye sredstva i sistemy upravleniya* [Electronic Means and Management Systems], 2015, no. 1–2, pp. 104–108.

13. Ilyashenko A. S., Laboshin L. Yu., Lukashin A. A. Sistema upravleniya raspredelennym kollektivom robotov [Management system for the distributed collective of robots]. *Ekstremalnaya robototekhnika* [Extreme Robotics], 2015, no. 1 (1), pp. 320–323.

14. Isupova T. D. Razrabotka podsistemy tekhnicheskogo zreniya robota androidnogo tipa kak obekta dvoynogo naznacheniya [Development of a subsystem of technical sight of the robot of androidny type as object of dual purpose]. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskije nauki* [Proceedings of the Tula State University. Engineering], 2015, no. 11–2, pp. 209–215.

15. Kabak I. S. Matematicheskaya model dlya prognozirovaniya nadezhnosti programmogo obespecheniya [The mathematical model for prediction of reliability of the software]. *Vestnik MGTU* [Bulletin of the MSTU], 2014, no. 1 (28), pp. 123–126.

16. Kaysner E., Raffo D., Vunsh-Vinsent S. Robototekhnika: proryvnye tekhnologii, innovatsii, intellektualnaya sobstvennost [Robototekhnika: disruptive technologies, innovations, intellectual property]. *Forsayt* [Forsythe], 2016, vol. 10, no. 2, pp. 7–27.
17. Klimov M. A., Vorotnikov S. A., Vybornov N. A. Sposob kalibrovki sistem lokalnoy navigatsii mobilnykh robotov [System calibration method of local navigation of mobile robots]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2017, no. 1, pp. 106–115.
18. Kozhemyakin I. V., Putintsev I. A., Semenov N. N., Chemodanov M. N. Razrabotka podvodnogo robototekhnicheskogo kompleksa, s ispolzovaniem otkrytykh sredstv modelirovaniya dvizheniya, dopolnennykh modelu gidroakusticheskogo vzaimodeystviya [Development of an underwater robotic complex, with use of the open simulators of movement added by model of hydroacoustic interaction]. *Izvestiya YuFU. Tekhnicheskie nauki* [Proceedings of the SFU. Engineering], 2016, no. 1 (174), pp. 88–101.
19. Kolker A. B., Livenets D. A., Kosheleva A. I., Zhmud V. A. Issledovanie variantov sozdaniya intellektualnykh sistem robototekhniki na baze odnoplatnykh kompyuterov i svobodnykh operatsionnykh sistem [Investigation of options of creation of intellectual systems of robotic technology on the basis of single board computers and the free operating systems]. *Avtomatika i programmnaya inzheneriya* [Automatic Equipment and Program Engineering], 2012, no. 1 (1), pp. 84–98.
20. Konyshov D. V., Vorotnikov S. A., Vybornov N. A. Upravlenie mimicheskimi apparatami servisnykh robotov pri sinteze emotsiy. [Control of mimic apparatus of service robots for emotions synthesis]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 3, pp. 216–229 ([http://hi-tech.asu.edu.ru/files/3\(27\)/216-229.pdf](http://hi-tech.asu.edu.ru/files/3(27)/216-229.pdf)).
21. Kostyukova A. V., Pikina Ye. V., Chibinev K. N., Shmatko M. Ye. Osnovnye elementy blok-skhemy programmnogo obespecheniya robota-gornospasatelya [Basic elements of the flowchart of the robot mine rescuer]. *Gornyy informatsionno-analiticheskiy byulleten* [Mountain Information and Analytical Bulletin], 2016, no. 3, pp. 384–388.
22. Larkin Ye. V., Kotov V. V., Kotova N. A. Otsenka effektivnosti programmnogo obespecheniya robota s ispolzovaniem setey Petri – Markova [An assessment of efficiency of the software of the robot with use of networks of Petri – Markov]. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki* [Proceedings of the Tula State University. Engineering], 2013, no. 9–2, pp. 156–162.
23. Medvedeva S. I., Shepelev S. O., Nekrasov A. A. Robototekhnicheskie sorevnovaniya kak mekhanizm proforientatsii podrostkov i obucheniya spetsialistov [Robototekhnicheskie sorevnovaniya kak mekhanizm proforientatsii podrostkov i obucheniya spetsialistov] [Robototekhnicheskie sorevnovaniya kak mekhanizm proforientatsii podrostkov i obucheniya spetsialistov]. *Innovatsionnye podkhody k resheniyu tekhniko-ekonomicheskikh problem* [Innovative Approaches to the Solution of Technical and Economic Problems], 2015, pp. 227–229.
24. Mikhaylova U. V., Mikhaylov Ye. A., Sarvarov A. S. Programmnye resheniya dlya razrabotki arkhitektury sistem upravleniya robotom [Software solutions for development of a system architecture of control of the robot]. *Elektrotekhnicheskie sistemy i komplekсы* [Electrotechnical Systems and Complexes], 2013, no. 21, pp. 111–117.
25. Nguen Tuan Zung, Shcherbatov I. A. Raspoznavanie obektov v sisteme tekhnicheskogo zreniya mobilnogo robota : ispolzovanie biblioteki FLANN I ALGORITMA SURF [Objects recognition in machine vision system of the mobile robot: the using of library “FLANN” and algorithm “SURF”]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 4, pp. 65–76 ([http://hi-tech.asu.edu.ru/files/4\(28\)/65-76.pdf](http://hi-tech.asu.edu.ru/files/4(28)/65-76.pdf)).
26. Otradnov K. K. Postroenie intellektualnogo programmnogo obespecheniya dlya upravleniya dvunogim shagayushchim robotom s postupatelnyimi kinematicheskimi parami v sustavakh nog [Creation of the intellectual software for control of the biped legged robot with progressive kinematic couples in joints of legs]. *Izvestiya vysshikh uchebnykh zavedeniy. Mashinostroenie* [Proceedings of the Higher Educational Institutions. Mechanical Engineering], 2012, no. 11, pp. 42–45.
27. Petrina A. M. O vzaimodeystvii intellektualnykh robotov, programmnogo obespecheniya i kompyuterov [About interaction of intelligent robots, the software and computers]. *Nauchno-tekhnicheskaya informatsiya. Seriya 2: Informatsionnye protsessy i sistemy* [Scientific and Technical Information. Series 2: Information Processes and Systems], 2015, no. 2, pp. 31–38.
28. Polyanskiy V. I. Problemnnye voprosy upravleniya riskami pri razrabotke, ispytaniyakh i prodenii resursa slozhnykh tekhnicheskikh sistem voennogo naznacheniya [Problematic issues of risk management by development, tests and extension of a resource of military difficult technical systems]. *Problemy informatiki* [Informatics Problems], 2012, no. 4 (16), pp. 89–92.
29. Rogatkin D. A., Lapaeva L. G. Mediko-tekhnicheskie trebovaniya k avtonomnym mobilnym servisnym meditsinskim robotam [Medical and technical requirements to independent mobile service medical robots]. *Robototekhnika i tekhnicheskaya kibernetika* [Robototekhnika and an Engineering Cybernetics], 2016, no. 2 (11), pp. 45–51.
30. Serov A. N., Savchenko Yu. V., Shipatov A. V., Sotnikov A. V. Programmno-apparatnyy kompleks kontrolya i otladki programmnogo obespecheniya vychislitel'nogo ustroystva dlya nazemnogo mobilnogo robota [A hardware and software system of monitoring and debugging of the software of the computing device for the terrestrial mobile robot]. *Izvestiya vysshikh uchebnykh zavedeniy. Elektronika* [Proceedings of the Universities. Electronics], 2014, no. 6 (110), pp. 43–50.
31. Starusev A. V. Metod otsenki i obespecheniya kachestva ispytaniy avtomatizirovannykh sistem [Method of the evaluation and quality assurance testing of automated systems]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2014, no. 4 (28), pp. 197–204 ([http://hi-tech.asu.edu.ru/files/4\(28\)/197-204.pdf](http://hi-tech.asu.edu.ru/files/4(28)/197-204.pdf)).
32. Starusev A. V., Lobeyko V. I., Litvinov S. P. Metod planirovaniya ispytaniy slozhnykh tekhnicheskikh sistem [Method of planning of tests of difficult technical systems]. *Izvestiya Volgogradskogo gosudarstvennogo tekhnicheskogo universiteta* [Proceedings of the Volgograd State Technical University], 2015, no. 2 (157), pp. 90–93.
33. Shmatko M. Ye., Pikina Ye. V., Chibinev N. N., Chibinev K. N. Osnovnye trebovaniya k programmnomu obespecheniyu robota-gornospasatelya [Main software requirements of the robot mine rescuer]. *Gornyy informatsionno-analiticheskiy byulleten* [Mountain Information and Analytical Bulletin], 2016, no. 1, pp. 135–138.
34. Uchaev D. Yu., Brumshteyn Yu. M., Azhmukhadedov I. M., Knyazeva O. M., Dyudikov I. A. Analiz i upravlenie riskami, svyazannymi s informatsionnym obespecheniem cheloveko-mashinnykh ASU tekhnologicheskimi protsessami v realnom vremeni [Analysis and management of risks, concerned with information support of human-machine automated control systems for technological processes operating in real time]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Control and High Technologies], 2016, no. 2 (34), pp. 82–97 ([http://hi-tech.asu.edu.ru/files/2\(34\)/82-97.pdf](http://hi-tech.asu.edu.ru/files/2(34)/82-97.pdf)).

35. Chernikov B. V. *Upravlenie kachestvom programmnoy obespecheniya* [Software quality management], Moscow, FORUM : INFRA-M Publ., 2012. 240 p.
36. Betty H. C. Cheng and 28 more. Software Engineering for Self-Adaptive Systems: A Research Roadmap. *Lecture Notes in Computer Science*, 2009, vol. 5525, pp. 1–26.
37. Medvidovic Nenand and others. Engineering Heterogeneous Robotics Systems: A Software Architecture-Based Approach. *Computer*, May 2011, vol. 44, issue 5, pp. 62–71.
38. Schlegel C., Lotz A., Lutz M., Berns K., Dillmann R., Maehle E. Model-driven software systems engineering in robotics: Covering the complete life-cycle of a robot. Special Issue: Robot Control Architectures. *Information Technology*, 2015, vol. 57 (2), pp. 85–98. DOI: 10.1515/itit-2014-1069.
39. Zimin G. A., Mordvinov D. A. Visual dataflow language for educational robots programming. *Trudy Instituta sistemnoy programmirovaniya RAN* [Proceedings of the Institute of System Programming, RAS], 2016, vol. 28, no. 2, pp. 45–62.
40. Yim Mark and others. Modular Self-Reconfigurable Robot Systems (Grand Challenges of Robotics). *IEEE Robotics & Automation Magazine*, March 2007, vol. 14, issue 1, pp. 43–52.

УДК 004.891.2

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ДЛЯ ОБЕСПЕЧЕНИЯ АВТОМАТИЗИРОВАННОГО УПРАВЛЕНИЯ ПРОЦЕССОМ КАТАЛИТИЧЕСКОГО РИФОРМИНГА В УСЛОВИЯХ НЕОПРЕДЕЛЕННОСТИ

Статья поступила в редакцию 16.04.2017, в окончательном варианте – 20.06.2017.

Джамбеков Азамат Матифулаевич, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, аспирант, ORCID 0000-0002-8768-9474, e-mail: azamat-121@mail.ru.

Для поддержки принятия решений по управлению процессом каталитического риформинга (КР) введен обобщенный критерий оптимальности (ОКО), включающий затраты на производство продукта и его октановое число. Описана задача управления процессом КР, которая заключается в определении таких управляющих воздействий, при которых обеспечивается экстремум ОКО. Разработана методика управления процессом КР в условиях неопределенности. Определены нечеткие цель и ограничения при управлении процессом КР. Нечеткая цель сформулирована как «ОКО должен быть минимальным». Одно из нечетких ограничений сформулировано как «Активность катализатора должна быть выше средней», а другое – «Состояние печи риформинга должно быть лучше среднего». Нечеткие цель и ограничения представлены в виде нечетких множеств на универсальном множестве альтернатив. В качестве альтернатив используются значения температуры на выходе из печи. Представлено выражение для определения нечеткого решения с использованием схемы Беллмана – Заде. На основе этого выражения определена альтернатива, при которой функция принадлежности нечеткого решения достигает максимума. Для полученной альтернативы с использованием гибридной математической модели процесса КР определены управляющие воздействия, обеспечивающие экстремум ОКО. Разработано программное обеспечение (ПО), которое реализует методику динамического управления процессом КР на ЭВМ в условиях неопределенности. Для обеспечения интерфейса с пользователем это ПО содержит ряд вкладок. Вкладка «Исходные данные» ПО необходима для заполнения значений зависимости обобщенного критерия оптимальности от температуры на выходе из печи, а также для задания функций принадлежности терм-множеств лингвистических переменных «Активность катализатора», «Состояние печи». Вкладка «Определение нечеткого решения» служит для задания нечеткой цели и нечетких ограничений с учетом заданных терм-множеств во вкладке «Исходные данные». Вкладка «Определение нечеткого решения» необходима для построения графика функций принадлежности нечеткого решения и определения альтернативы, соответствующей максимуму данной функции принадлежности. Вкладка «Определение вектора управлений» служит для определения управляющих воздействий, соответствующих экстремуму ОКО.

Ключевые слова: каталитический риформинг, обобщенный критерий оптимальности, нечеткая цель, нечеткое ограничение, активность катализатора, состояние печи риформинга, температура на выходе из печи риформинга, универсальное множество альтернатив, функция принадлежности, лингвистическая переменная

USE OF INFORMATION TECHNOLOGIES FOR PROVIDING AUTOMATED CONTROL OF THE PROCESS OF CATALYTIC RIFORMING IN THE CONDITIONS OF UNCERTAINTY

The article has been received by editorial board 16.04.2017, in the final version – 20.06.2017.

Dzhambekov Azamat M., Astrakhan State Technical University, 16 Tatishchev St., Astrakhan, 414056, Russian Federation, post-graduate student, ORCID 0000-0002-8768-9474, e-mail: azamat-121@mail.ru.

To support the decision-making on the management of the catalytic reforming process (CR), a generalized optimality criterion (GOC) is introduced, including the cost of producing the product and its octane number. The task of controlling the process of the CR is described, which consists in determining such control actions under which the extremum of the GOC is provided. A method for controlling the process of the CR under conditions of uncertainty has been developed. The fuzzy goal and limitations in the management of the CR process have been determined. The fuzzy goal is formulated as “The