

## ОНТОЛОГИЧЕСКИЙ АНАЛИЗ ДЕФЕКТОВ ПРИ ПРОЕКТИРОВАНИИ КОМПОНЕНТОВ АППАРАТНО-ПРОГРАММНЫХ КОМПЛЕКСОВ

В.Е. Гвоздев<sup>1</sup>, Д.В. Блинова<sup>2</sup>

Уфимский государственный авиационный технический университет, Уфа, Россия

<sup>1</sup>wega55@mail.ru, <sup>2</sup>blinova.darya@gmail.com

### Аннотация

Статья посвящена анализу дефектов в аппаратно-программных комплексах (АПК) как составной части управления функциональной безопасностью сложных систем. Выделены места возникновения непреднамеренных дефектов, способных нарушить функциональную безопасность АПК. Предложен подход к рассмотрению понятия «дефект» как многомерной системы, описаны её проекции: влияние на правообладателей, технологическая, управленческая. Приведена классификация дефектов, характерных для программных продуктов и проектов. Предложены системные и математические модели программных продуктов и проектов как основа выявления возможных мест возникновения дефектов. Описана роль субъективной составляющей при разработке компонентов АПК.

**Ключевые слова:** функциональная безопасность, дефект аппаратно-программного комплекса, классификация дефектов, холистический подход, программный продукт, программный проект.

### Введение

В литературе, посвященной управлению сложными системами [1, 2] отмечается необходимость исследования не только «полезных» (useful), но и «вредных» (harmful) функций системы. В работе [3] подчёркивается необходимость смещения акцентов в проблематике создания аппаратно-программных комплексов (АПК) от вопросов штатной эксплуатации к вопросам их безопасного функционирования. Там же отмечается необходимость совершенствования технологий разработки, позволяющих обеспечивать защиту изделий как от злонамеренных действий, так и от непреднамеренных ошибок, допускаемых разработчиками на разных стадиях жизненного цикла АПК. В работах [3-5] отмечается, что одним из перспективных современных направлений исследований в области системной инженерии является создание методологических и теоретических основ дефектологии АПК. При этом следует выделять классы задач, связанные с обеспечением технологической и эксплуатационной безопасности АПК. Предлагаемая классификация является условной и основана на задачах, выделенных при анализе литературных источников [3, 5]:

- задачи, решаемые в рамках комплексной проблемы защиты информации [3], в том числе задачи, связанные с обеспечением конфиденциальности, целостности и доступности информации для случая, когда в решении задач активное участие принимает пользователь [5];
- задачи, связанные с построением АПК для управления и обработки информации в реальном времени при относительно малом участии пользователей (задачи функциональной безопасности) [5]. Под функциональной безопасностью АПК понимается свойство АПК сохранять работоспособность в соответствии со своим назначением при случайных дестабилизирующих воздействиях и отсутствии злоумышленного влияния на программную, аппаратную составляющую или базы данных).

В данной работе авторами под дефектом понимается предпосылка к ошибке, возникновение (реализация) которой влечет за собой значимые потери для пользователя АПК. Следует обратить внимание на то, что к настоящему времени не сформировано единой и устойчивой терминологии в области дефектологии; например, в работах [5-7] описана проблематика классификации дефектов и затрагиваются вопросы выработки понятий по данному разделу.

Проведённый анализ литературы позволяет сделать заключение о необходимости различать подходы к управлению дефектами, вносимыми в изделия преднамеренно, и дефектами, допускаемыми разработчиками АПК непреднамеренно, при том, что масштабы негативных последствий от непреднамеренных дефектов могут многократно превосходить масштабы последствий злонамеренных действий. Кроме того, следует по-разному подходить к управлению непреднамеренно допускаемыми дефектами на разных стадиях жизненного цикла изделий в силу различной природы дефектов.

По нашему мнению, в литературе [8, 9], посвящённой проектированию, обеспечению эксплуатации и развитию систем обработки и обмена информацией, в основном, внимание акцентируется на полезных функциях. В то же время, понятие «дефект» необходимо увязывать с понятием «вредная функция». Изучению вредных функций не уделялось должного внимания в исследованиях, связанных с управлением качеством систем информационной поддержки управления. К настоящему времени получила развитие методологическая, модельная и инструментальная основа управления надёжностью изделий электронной техники, но аналогичное заключение нельзя сделать относительно надёжности программной составляющей АПК, хотя известны публикации [10] по тематике инженерии надёжности программных систем (Software Reliability Engineering).

Анализ литературных источников, а также опыт участия в реализации программных проектов, позволяет сделать заключение об отсутствии чётких методических основ назначения норм надёжности программных продуктов и функциональной безопасности программных систем. Требуется развития методологическая, методическая и модельная, инструментальная основа проектирования программных систем, продуктов и модулей, удовлетворяющих заданным требованиям по надёжности и функциональной безопасности; способов доказательства соответствия фактических свойств изделий заданным требованиям. Важность подобных исследований подтверждается, например, данными, приведёнными в [11]. В этом источнике приводятся сведения о потерях (финансовых, временных), обусловленных дефектами в прикладных программных системах, о количестве программных продуктов, претерпевающих переработку из-за выявленных дефектов.

Необходимо подчеркнуть влияние дефектов в организации программных проектов на наличие дефектов в программных продуктах. Программный проект есть развернутый во времени программный продукт. Различные фазы жизненного цикла программного проекта представляют собой последовательное преобразование ожиданий правообладателей в машинные коды.[12]. Из этого можно заключить, что дефекты организации программных проектов являются первичными по отношению к дефектам в программных продуктах.

Многомерность понятия «дефект» в проекте диктует его исследование, по крайней мере, в следующих проекциях.

- *Социальная.* Необходимость исследования этой проекции обусловлена тем, что дефекты в АПК способствуют реализации вредных функций системой, в которую они встроены. Это, в свою очередь, может представлять угрозу для внешних и внутренних правообладателей системы [14]. Иными словами, понятие «дефект» напрямую связано с понятиями «угроза» и «риск» [5] (угроза обусловлена воздействием внешней среды на проект, риск ассоциирован с внутренней средой объекта управления).

- *Технологическая.* Необходимость этой проекции обусловлена тем, что причинами дефектов являются:
  - ограниченные возможности изучения потребностей, ожиданий и желаний правообладателей [13] (внешних и внутренних [14]). Это обусловлено слабой формализацией известных технологий извлечения реальных потребностей пользователей;
  - ограниченность методов, моделей и инструментов планирования и управления проектами в условиях высокой изменчивости как внешней, так и внутренней сред проектов, связанных с созданием, модернизацией и развитием АПК [11, 15-17]. Центральной составляющей внешней среды являются потребности, желания, ожидания правообладателей, а также ограничения, накладываемые на способы реализации программных проектов в виде бюджета и сроков реализации проекта. Внутренняя среда представляет собой совокупность основных, вспомогательных и обеспечивающих процессов [18], результатом реализации которых является АПК;
  - отсутствие «бесшовных» технологий реализации как отдельного проекта [19], так и при построении интегрированных информационно-коммуникационных систем на базе локальных АПК [20].
  - индивидуальные пристрастия разработчиков в выборе инструментария реализации проектов (в литературе этому соответствует понятие «инструментальный ящик проектов» [19]). Интеграция компонентов, реализованных разным набором инструментов, также является причиной возникновения дефектов.
- *Управленческая.* Организационная структура, методология, методическое и другие виды обеспечения зависят от масштаба и сложности проекта. Источниками возникновения дефектов могут служить, например, модель жизненного цикла (при реализации IT-проектов в условиях высокой изменчивости требований организационным дефектом будет выбор «водопадной» модели жизненного цикла; альтернативой широко применяемой «водопадной» модели в таких условиях являются подходы, соответствующие Agile process [15]), либо неадекватный набор нормативных документов, положенных в основу реализации проекта [8]. Следует особо подчеркнуть роль субъективной составляющей в реализации IT-проектов как источника дефектов. Многогранность субъективной составляющей подтверждается также статистическими данными, показывающими доминирующую роль субъективной составляющей IT-проектов на исход проекта [11, 15].

Дефекты (с точки зрения потребителей информационных продуктов и услуг) могут возникать и в изначально функционально-пригодных АПК. Это обусловлено содержанием понятия «отказ» применительно к программной составляющей АПК [10]. Отказом считается отставание в темпе предоставления информации от того, который необходим для обеспечения эффективного управления системой, в которую встроено АПК. Причиной этого является то, что внешняя среда, в которой фактически оказывается управляемая система, с течением времени начинает отличаться от той, описание которой содержится в техническом задании на разработку АПК.

Отмеченные обстоятельства позволяют сделать заключение о необходимости активизации исследований в области дефектологии АПК, что создаст базу для обеспечения требуемого уровня функциональной безопасности систем информационной поддержки управления.

## 1 Классификация дефектов

Многомерность и динамическая природа объекта исследования «*дефект программного компонента АПК*» диктует множественность подходов к классификации дефектов.

В общем случае целью решения задачи классификации является выделение относительно устойчивых образований («классов дефектов»), что служит основанием развития теоретических положений и разработки инструментальных средств для исследования дефектов с учётом различия приоритетов, соответствующих разным классам.

В первую очередь следует разграничить дефекты в продуктах и дефекты в проектах. Дефекты продуктов являются производными дефектов проектов. Коренной причиной дефектов обоих классов является неопределённость как в требованиях к потребительским свойствам продукта, так и в способах обеспечения этих требований [21].

Заметим, что в [22] считается, что на вход программного проекта поступают требования пользователей (функциональные, иначе называемые «требования-возможности», описывающие, решение каких классов задач должен обеспечивать продукт; нефункциональные, иначе «требования-ограничения», определяющие такие характеристики, как надёжность, быстродействие и др.)<sup>1</sup>. Это подчёркивает сложность формирования спецификации требований пользователей, а также иллюстрирует стремление разработчиков методологий реализации ИТ-проектов дистанцироваться от возможной неудачной реализации проекта.

Если в основу классификации продуктов положить показатель сложности (для программного модуля и приложения, построенного по модульному принципу, определён формальный показатель в виде цикломатического показателя сложности [17]), то можно выделить следующие классы объектов:

- программный модуль/компонента АПК, предназначенный для реализации ограниченного числа операций в рамках реализации функции, представляющей ценность для конечного пользователя;
- локальная программная система, предназначенная для реализации ограниченного числа функций, представляющих ценность для пользователя;
- компонента функциональной подсистемы, решающая ограниченный круг задач, связанных со сбором, обеспечением доступностью, систематизацией, обеспечением сопоставимости, обеспечением хранения, обработки и представления информации в рамках широкого круга задач, связанных с информационным обслуживанием выделенной целевой группы внешних правообладателей [14, 23, 24];
- функциональная подсистема, решающая широкий круг задач, связанных с информационным обслуживанием выделенной целевой группы внешних правообладателей;
- корпоративная информационная система, являющаяся исчерпывающей информационной моделью организации;
- информационно-коммуникативная система, создаваемая на базе информационно-вычислительных систем и обеспечивающая взаимодействие и информационное обслуживание территориально-распределённых агентов [20].

Различное содержание понятия «дефект» в зависимости от уровня сложности объекта связано с использованием терминов: «bugs», «defect», «fault». Первый термин применим на уровне модуля (точка зрения разработчика); второй – на уровне локальной программной системы (точка зрения разработчика); третий – на уровне подсистемы /системы (точка зрения пользователя).

Каждому из этих классов объектов соответствует свой подход к исследованию дефектов и их проявлений. Примерами исследований, ориентированных на управление дефектами и соответствующих некоторым из выделенных классов, служат:

<sup>1</sup> Согласно IEEE Recommended Practice for Software Requirements Specifications IEEE Std 830-1998 требования должны обладать признаками корректности, однозначности, полноты, согласованности, верифицируемости, модифицируемости, трассируемости и ранжированные по важности.

- на уровне модуля – прогнозирование количества дефектов на основе показателя сложности [25];
- на уровне локальной программной системы – модели надёжности [10, 26, 27];
- модели для оценки надёжности функционально-сложных систем со сложными стратегиями ремонта и технического обслуживания – марковский анализ [28].

Следует подчеркнуть, что приведённая классификация, как и всякая классификация, является условной. Это обуславливает изменение содержания понятий «операция», «функция», «компонента», «система» по мере развития интеллектуальности средств сбора, обработки, хранения и представления информации, и, следовательно, изменяет содержание понятий «дефект», «ошибка», «отказ».

Классификация программных проектов по показателю сложности возможна по аналогии с СММИ [29]:

- проекты, реализация которых возможна без использования систематических процедур, по интуиции (*ad hoc*);
- проекты, основанные на структуризации процессов;
- проекты, основанные на использовании полезных практик, закреплённых в нормативных документах, руководствах и рекомендациях;
- проекты, основанные на использовании взаимосвязанной совокупности нормативных документов, моделей и инструментов («инструментального ящика проекта» [19]);
- проекты, основанные на комплексном формальном анализе альтернативных вариантов построения как структур взаимосвязанных процессов, так и способов реализации отдельных этапов этих процессов [19, 30].

Поэтапная реализация АПК позволяет рассматривать проекты и продукты как системы с конечным числом элементов или состояний системы [31].

Изложим содержание известной модели [31] системы  $\Sigma$  с конечным числом состояний:

$$(1) \quad \Sigma = (U, Y, Q, \lambda, \gamma),$$

в терминах программного проекта и продукта.

Для *продукта* составляющие модели получают следующее содержание:

$U$  – множество требований пользователей (внешний облик АПК/ограничения АПК);

$Y$  – множество потребительских свойств АПК;

$Q$  – множество состояний продукта (модель жизненного цикла продукта);

$\lambda$  – множество функций перехода между стадиями жизненного цикла:

$$(2) \quad \lambda_i: Q_i \times U_i \rightarrow Q_{i+1}, \quad i = \overline{1, n-1},$$

здесь  $i$  – номер стадии;

$\gamma$  – множество функций выхода (результатов  $i$ -й стадии, поступающих на вход  $(i+1)$ -й стадии):

$$(3) \quad \gamma_i: Q_i \times U_i \rightarrow Y_i.$$

Для *проекта* содержание модели получает следующее толкование:

$U$  – множество требований к проекту (сроки; стоимость; ограничения на условия реализации);

$Y$  – множество характеристик качества проекта (социальных; политических; финансовых; технологических; научных; образовательных);

$Q$  – множество компонентов основного (реализации проекта), вспомогательного (инструменты), обеспечивающего (управление) процессов;

$\lambda$  – множество работ, реализуемых на  $j$ -й стадии проекта:

$$(4) \quad \lambda_j: Q_j \times U_j \rightarrow Q_{j+1}, \quad j = \overline{1, n-1},$$

$\gamma$  – множество промежуточных продуктов, получаемых на стадиях проекта (критериев перехода от  $j$ -й стадии к  $(j+1)$ -й стадии):

$$(5) \quad \gamma_j: Q_j \times U_j \rightarrow Y_{j+1}.$$

Приведённые системные модели служат основанием для выделения возможных мест возникновения дефектов. Для продуктов: требования; потребительские свойства; модель жизненного цикла; переходы между стадиями жизненного цикла; входные/выходные результаты стадий жизненного цикла. Для проектов: требования; показатели ценности результатов проекта с точки зрения разных правообладателей; состав процессов; состав работ; качество промежуточных продуктов.

Основу классификации мест возникновения дефектов в продукте может составить структура типового процесса разработки и использования АПК (рисунок 1). Предлагаемая модель построена по аналогии с известной V-моделью жизненного цикла программного продукта [32].

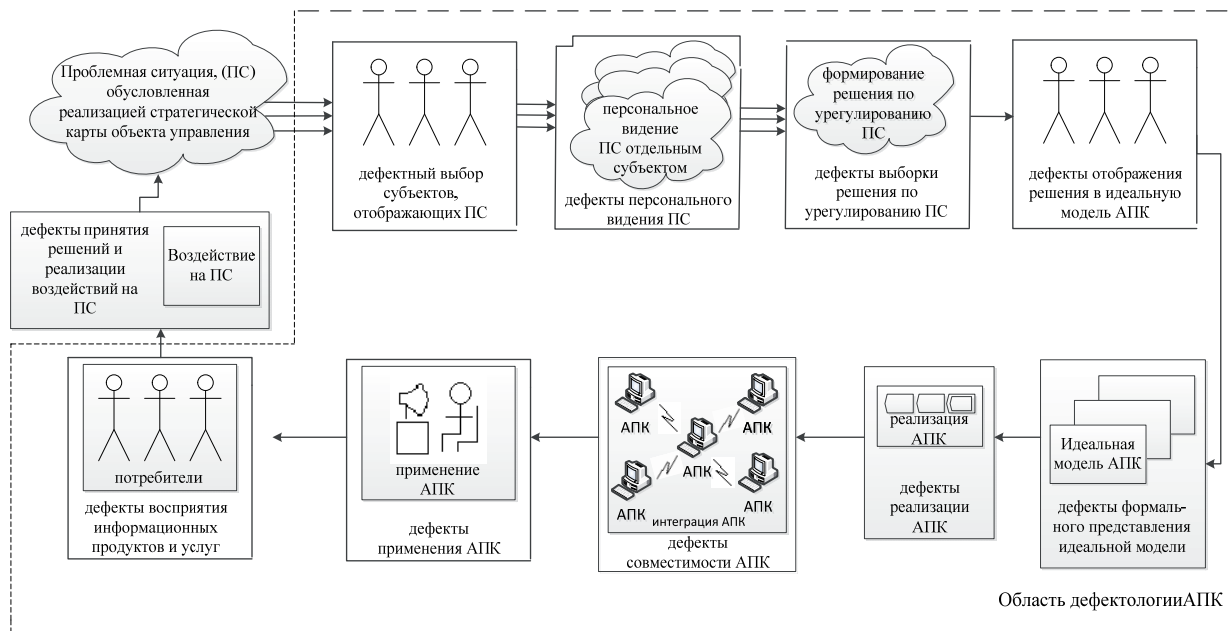


Рисунок 1 – Места возникновения дефектов при создании и использовании АПК

Использование такой классификации поможет снизить неопределённость при разработке программного проекта за счет структуризации процесса. Следует подчеркнуть, что в литературе широко освещаются проблемы возникновения дефектов на стадии реализации (кодирования и испытания), но причинам возникновения дефектов на других стадиях не уделяется должного внимания.

Основу классификации мест возникновения дефектов в программном проекте также могут составить ключевые направления совершенствования IT-проектов, определённые в работах [15, 16].

Предлагаемые подходы к классификации дефектов в компонентах АПК и проектах реализации их компонент не могут считаться исчерпывающими. Приведённые примеры лишь иллюстрируют множественность подходов к изучению дефектов.

Классификацию моделей дефектов можно увязать с классификацией их типов, например:

- конструктивные;

- технологические;
- временные.

Примером математической модели дефектов в классе моделей конструктивных дефектов является установление формальных соотношений между показателем сложности продукта и ожидаемым числом дефектов [25].

Примером структурной модели, отражающей технологические аспекты предупреждения и ранней идентификации дефектов, является V-модель жизненного цикла программного продукта [32].

Примером модели, ориентированной на сокращении времени разработки программной компоненты АПК с требуемыми потребительскими свойствами в условиях высокой изменчивости требований пользователей, является Agile Process [15].

В качестве примера классификации моделей дефектов в проектах можно привести следующее:

- моделирование последствий дефектов во внешней и внутренней средах проекта;
- моделирование дефектов в организации проектов;
- моделирование последствий дефектов в планах проектов.

Проведённый анализ литературных источников не позволил выявить модели дефектов, относящихся к выделенным классам.

## 2 Холистический подход к анализу дефектов

Одним из фундаментальных подходов к исследованию сложных систем является рассмотрение её как взаимодействующей совокупности целостностей [31, 33]. Анализ критических факторов успеха и неудач, связанных с реализацией IT-проектов [15, 16], позволяет сделать однозначный вывод о решающей роли субъективной составляющей на результаты проекта. Рассуждая о дефектах, следует подчеркнуть, что различные правообладатели по-разному относятся к одним и тем же свойствам АПК, входящим в состав сложных систем, в том числе к наличию и проявлению разных дефектов. Это обусловлено различием точек зрения, в рамках которых каждый правообладатель относится к ценностям, создаваемым АПК [9, 24].

Известно, что каждую систему следует рассматривать с позиции внешнего поведения и внутреннего устройства. Внешнее поведение – это основа оценивания системы потребителем. Внутреннее устройство – это основа оценивания системы и её компонентов разработчиками. Различие точек зрения на систему приводит к заключению о различном отношении к одним и тем же дефектам разных правообладателей. Рассмотрение дефекта как неотъемлемого свойства сложных систем, с одной стороны, различное видение разными правообладателями одной и той же системы, с другой стороны, определяющая роль субъекта в разработке компонентов системы и оценивании их потребительских свойств, с третьей, позволяют сделать заключение о множественности холонических моделей дефектов. На рисунках 2-4 в качестве примера приведены холонические модели дефектов, соответствующие рассмотрению компонентов АПК внешними и внутренними правообладателями.

Дефекты характерны каждому из отношений и сущностей, каждой сущности соответствует дефект своего класса. Построение моделей связей дефектов видится авторами актуальной задачей, требующей дальнейшего рассмотрения.

Приведённые примеры призваны подчеркнуть такую грань холистического подхода, как единство объективного и субъективного (на эту особенность обращается внимание, например, в [34] со ссылкой на Я. Сметса). Актор (субъективная компонента) является фокусом управления функциональной безопасности системы: он выделяет внешние объекты, для информационной поддержки управления которыми используется АПК. Он определяет страте-

гию, критерии качества и ограничения управления. Актор формирует внешний облик АПК, определяет стратегию построения, использования и развития АПК.

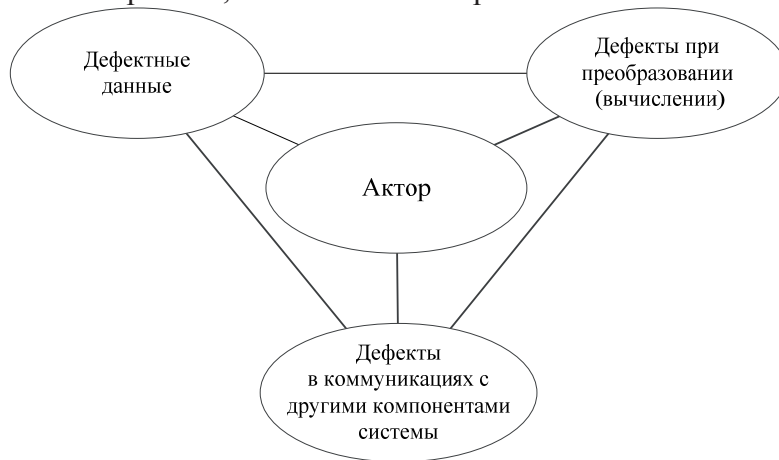


Рисунок 2 – Холоническая модель дефекта с точки зрения создания и обеспечения функционирования и развития АПК



Рисунок 3 – Холоническая модель дефекта с точки зрения пользователя АПК

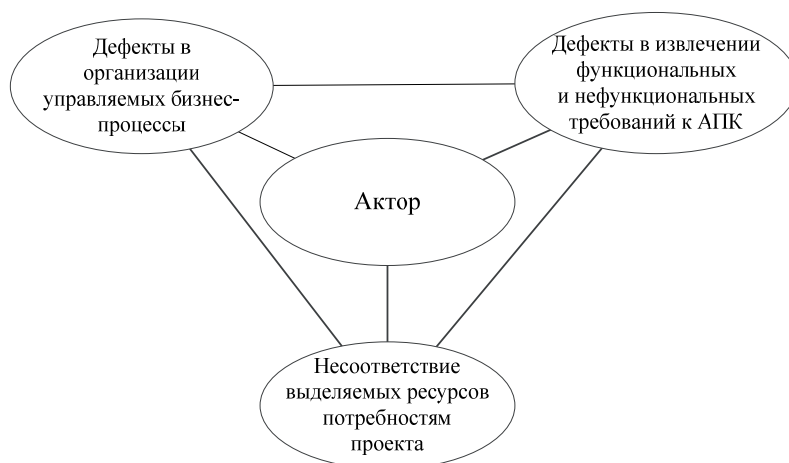


Рисунок 4 – Холоническая модель дефекта с точки зрения руководителя проекта



Следует отметить, что содержание компонентов представленных холонов (и, следовательно, содержание ассоциированных с ними дефектов) зависит от уровня рассмотрения АПК: программного модуля/технического устройства, локальной автоматизированной системы; компоненты функциональной подсистемы корпоративной информационной системы; функциональной подсистемы корпоративной информационной системы; корпоративной информационной системы; территориально распределённой информационно-коммуникационной системы.

## Заключение

Рассмотрены подходы к классификации дефектов в компонентах АПК и проектах реализации их компонент. Изложено описание общей модели сложных систем в терминах продуктов (компонентах АПК) и проектах их реализации.

Обосновывается положение о плодотворности холистического подхода к изучению дефектов в компонентах АПК с позиций системного анализа. Это обусловлено неразрывным единством объективной (ограниченность инструментальных средств проектирования; ограниченная область применения системных, знаковых моделей и т.д.) и субъективной (различное отношение разных правообладателей к одним и тем же свойствам изделий) точек зрения.

Приведён ряд холонических моделей дефектов, отражающих разные точки зрения правообладателей на дефекты и их проявления.

## Список источников

- [1] *Лапыгин Ю.Н.* Стратегический менеджмент. Учебное пособие – М.: Высшее образование, 2007. – 174 с
- [2] *Chai, K.-H.* A TRIZ-Based Method for New Service Design. National University of Singapore / Kan-Hin Chai, Jun Zhang, Kay-Chuan Tan // Journal of Service Research. – 2005. – Vol. 8, No. 1. – P. 48-66.
- [3] *Нагибин, С.Я.* Технологическая безопасность программного обеспечения – новая проблема в области создания информационных систем / С.Я. Нагибин, Б.П. Пальчун, Л.М. Ухлинов // Информационное общество. – 1995. - Вып. 6. - с. 45-49.
- [4] *Бородакий, Ю.В.* Проблема имитационного моделирования дефектоскопических свойств компьютерной инфосферы / Ю.В. Бородакий, Р.М. Юсупов, Б.П. Пальчун // Труды 3-й всероссийской научно-практической конференции «Имитационное моделирование. Теория и практика». - Санкт-Петербург, 2007. - с. 87-92.
- [5] *Липаев, В.В.* Функциональная безопасность программных средств / В.В. Липаев. - М.: СИНТЕГ, 2004. – 348 с.
- [6] *Марков, А.С.* Систематика уязвимостей и дефектов безопасности программных ресурсов / А.С. Марков, А.А. Фадин // Защита информации INSIDE. - 2013. - №3. – с. 2-7.
- [7] *Черняховская, Л.Р.* Разработка моделей и методов интеллектуальной поддержки принятия решений на основе онтологии организационного управления программными проектами/ Л.Р. Черняховская, А.И. Малахова //Онтология проектирования, № 4 (10), 2013. – с. 42-52.
- [8] *Липаев В.В.* Процессы и стандарты жизненного цикла сложных программных средств. М.: СИНТЕГ, 2006. – 260 с.
- [9] Учебник 4СЮ. Версия 1.0. М.: 4СЮ, 2011.– 228 с.
- [10] *Липаев В.В.* Надежность программных средств. М.: СИНТЕГ, 1998. – 232 с.
- [11] CHAOS MANIFESTO 2014: Value versus Success & The Orthogonals. The Standish Group International, <http://blog.standishgroup.com/post/14>.
- [12] *Гвоздев, В.Е.* Пирамида программного проекта/В.Е. Гвоздев, Б.Г. Ильясов //Программная инженерия. Теоретический и прикладной научно-технический журнал. № 1, 2011. – с. 16-24.
- [13] *Christel, M.G.* Issues in Requirements Elicitation / M.G. Christel, K.C. Kang // Technical Report CMU/SEI – 92 – TR – 012 Esc – TRM – 92 – 012, 1992. [http://resources.sei.cmu.edu/asset\\_files/TechnicalReport/1992\\_005\\_001\\_16478.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalReport/1992_005_001_16478.pdf) (Актуально на 24.10.2015).

- [14] **Bourne L.** SRMM Stakeholder Relationship Management Maturity PMI Global Congress EMEA 2008 19-21 May 2008 St. Julians, Malta URL: [http://www.mosaicprojects.com.au/Resources\\_Papers\\_067.html](http://www.mosaicprojects.com.au/Resources_Papers_067.html) (Актуально на 24.10.2015).
- [15] CHAOS Manifesto 2011: The Laws of CHAOS and the CHAOS 100 Best PM Practices. The Standish Group International, Incorporated URL: <http://immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/S110415C.pdf> (Актуально на 05.10.2015).
- [16] CHAOS MANIFESTO 2013: Think Big, Act Small. The Standish Group International, Incorporated URL: <https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf> (Актуально на 05.10.2015).
- [17] ESA PSS-05-10 Guide to software verification and validation, March 1995 URL: <ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0510.pdf> (Актуально на 05.10.2015).
- [18] A Guide to the Project Management Body of Knowledge: (PMBOK Guide), Project Management Institute, 2010
- [19] **Милошевич Д.** Набор инструментов для управления проектами. М.: Компания АйТи: ДМК Пресс, 2008. – 729 с.
- [20] **Тимофеев А.В.** Адаптивное управление и интеллектуальный анализ информационных потоков компьютерных сетей. – СПб: ООО «Анатолия», 2012. – 280 с.
- [21] **Гвоздев, В.Е.** Системное и структурное моделирование требований к программным продуктам и проектам / В.Е. Гвоздев, Д.В. Блинова, Н.И. Ровнейко, О.П. Ямалова // Программная инженерия, 2013. – с. 2-10.
- [22] ESA PSS-05-0 European Space Agency software engineering standards, February 1991 URL: <ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS050.pdf> (Актуально на 05.10.2015).
- [23] ГОСТ Р ИСО/МЭК 15288 – 2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.
- [24] **Каплан, Р.С.** Сбалансированная система показателей: от стратегии к действию / Р.С. Каплан, Д.П. Нортон. - М.: ЗАО «Олимп-Бизнес», 2003. – 214 с.
- [25] **Гвоздев, В.Е.** Оценка количества дефектов в программных модулях на основе показателей сложности // В.Е. Гвоздев, А.С. Субхангулова, О.Я. Бежаева // Информационные технологии и системы (ИТиС-2015): Тр. 4-й Междунар. науч. конф. (2015 г., Банное, Россия). - с. 12-14.
- [26] **Луи М.Р.** Handbook of Software Reliability Engineering, IEEE and McGraw-Hill, 1996. – 850 p.
- [27] **Lee, D.Y.** Using Software Reliability Models for Security Assessment - Verification of Assumptions / D.Y. Lee, M. Vouk, L. Williams // ISSRE 2013 (4-7 Nov 2013, Pasadena). - P. 23-24.
- [28] ГОСТ Р 51901.15-2005 – Менеджмент риска. Применение марковских методов.
- [29] CMMI for Development, Version 1.3, Software Engineering Institute, 2010. <http://www.sei.cmu.edu/reports/10tr033.pdf> (Актуально на 24.06.2015).
- [30] IEEE Guide to the Software Engineering Body of Knowledge. SWEBOOK, 2004. <http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOOK.pdf> (Актуально на 24.06.2015).
- [31] **Касту Дж.** Большие системы. Связность, сложность и катастрофы: Пер. с англ., 1982. – 216 с.
- [32] **Гвоздев, В.Е.** Элементы системной инженерии: методологические основы разработки программных систем на основе V-модели жизненного цикла / В.Е. Гвоздев, М.Б. Гузаиров, Б.Г. Ильясов, О.Я. Бежаева - М.: Машиностроение, 2013. – 180 с.
- [33] **Виттих В.А.** Ситуационное управление с позиций постнеклассической науки/В.А. Виттих//Онтология проектирования. №2(4). 2012. – с.7-15.
- [34] **Сергеева Л.А.** Актуальность системного подхода в современном информационном обществе // Философские проблемы информационных технологий и киберпространства, № 2, 2011. - с. 248-254.

## ONTOLOGICAL ANALYSIS OF DEFECTS AT THE DESIGNING OF HARDWARE-SOFTWARE COMPLEXES

V.E. Gvozdev<sup>1</sup>, D.V. Blinova<sup>2</sup>

Ufa State Aviation Technical University, Ufa, Russia

<sup>1</sup>wega55@mail.ru, <sup>2</sup>blinova.darya@gmail.com

### Abstract

The article focuses on the analysis of defects in hardware-software complexes as an integral part of the functional safety management of complex systems. The analysis of unintentional defects locations that could affect the functional safety of hardware-software complexes is made. An approach to the concept of "defect" as a multidimensional system, de-

scribed its projections: impact on stakeholders, technological, managerial is proposed. The classification of characteristic defects for software products and projects are shown. A system and mathematical models of software products and projects as a basis to identify possible locations of defects is described. The role of the subjective component in the designing of the components of hardware-software complexes is demonstrated.

**Key words:** *function safety, defect of hardware-software complex, classification of defects, holistic approach, program project, program product.*

## References

- [1] **Lapigin Yu. N.** Strategicheskij menedgment [Strategic Management]: Uchebnoe posobie – M.: Vishee obrazovanie, 2007. – 174 p. (In Russian).
- [2] **Kan-Hin Chai, Jun Zhang, Kay-Chuan Tan** A TRIZ-Based Method for New Service Design. National University of Singapore// Journal of Service Research, Volume 8, №1, August 2005, pp. 48-66.
- [3] **Nagibin S.Ya., Palchun B.P., Uhlinov L.M.** Technologicheskaja bezopasnost programmnoho obespechenija – novaja problema v oblasti sozdaniya informacionnih sistem [Software technologic safety – new problem in information systems development area] // Informacionnoe obshestvo, 1995. Issue 6. - pp. 45-49. (In Russian).
- [4] **Borodakij Yu.V., Yusupov R.M., Palchun B.P.** Problema imitacionnogo modelirovanija defectoskopicheskikh svoystv komputernoj infosferi [Problem of imitation modelling of computing infosphere defectoscopy features] / Proceedings of the 3<sup>rd</sup> Russian conf. «Imitation modelling. Theory and Practice» Sankt-Peterburg, 2007. - pp. 87-92. (In Russian).
- [5] **Lipaev V.V.** Funkcionalnaja bezopasnost programmnh sredstv [Function safety of software]. – M.: SINTEG, 2004. – 348 p. (In Russian).
- [6] **Markov A.S., Fadin A.A.** Sistematika ujazvimostei i defektov bezopasnosti programmnh resursov [Taxonomy vulnerabilities and defects of software safety] // Zascshita informacii INSIDE №3, 2013. – pp. 2-7. (In Russian).
- [7] **Chernahovskaja L.R., Malahova A.I.** Development Of Intellectual Decision Support Models And Methods Based On Ontology Of Software Projects Organization Management // Ontology of Designing, № 4 (10), 2013. – pp. 42-52 . (In Russian).
- [8] **Lipaev V.V.** Processi I standarti zhiznennogo tsikla slozhnih programmnh sredstv [Processes and standarts of complicated software lifecycle]. – M.: SINTEG, 2006. – 260 p. (In Russian).
- [9] Uchebnik 4CIO. Versija 1.0. M.: 4CIO, 2011.– 228 p. (In Russian).
- [10] **Lipaev V.V.** Nadejnost programmnh sredstv [Reliability of software]. – M.: SINTEG, 1998. – 232 p. (In Russian).
- [11] CHAOS MANIFESTO 2014: Value versus Success & The Orthogonals. The Standish Group International, <http://blog.standishgroup.com/post/14>
- [12] **Gvozdev V.E., Ilyasov B.G.** Pyramid of software project // Programmnaia injenerija. Teoreticheskij i prikladnoi nauchno-tehnicheskij zhurnal., № 1, 2011 – pp. 16-24 (In Russian).
- [13] **Christel M.G., Kang K.C.** Issues in Requirements Elicitation: Technical Report CMU/SEI – 92 – TR – 012 Esc – TRM – 92 – 012, September 1992.  
URL: [http://resources.sei.cmu.edu/asset\\_files/TechnicalReport/1992\\_005\\_001\\_16478.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalReport/1992_005_001_16478.pdf) (Valid on: 24.10.2015)
- [14] **Bourne L.** SRMM Stakeholder Relationship Management Maturity PMI Global Congress EMEA 2008 19-21 May 2008 St. Julians, Malta URL: [http://www.mosaicprojects.com.au/Resources\\_Papers\\_067.html](http://www.mosaicprojects.com.au/Resources_Papers_067.html) (Valid on 24.10.2015)
- [15] CHAOS Manifesto 2011: The Laws of CHAOS and the CHAOS 100 Best PM Practices. The Standish Group International, Incorporated URL: <http://immagic.com/eLibrary/ARCHIVES/GENERAL/GENREF/S110415C.pdf> (Valid on 05.10.2015)
- [16] CHAOS MANIFESTO 2013: Think Big, Act Small. The Standish Group International, Incorporated URL: <https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf> (Valid on 05.10.2015)
- [17] ESA PSS-05-10 Guide to software verification and validation, March 1995 URL: <ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS0510.pdf> (Valid on 05.10.2015)
- [18] A Guide to the Project Management Body of Knowledge: (PMBOK Guide), Project Management Institute, 2010
- [19] **Milosevic D.Z.** Project Management ToolBox: Tools and Techniques for the Practicing Project Manager. – John Wiley & Sons, 2003. – 600 p.
- [20] **Timofeev A.V.** Adaptivnoe upravlenije I intellektualnij analiz informacionnih potokov v komputernih setjah [Adaptive control and intellectual analyses of information flows in networks] – SPb: OOO «Anatiliya», 2012. – 280 p. (In Russian).
- [21] **Gvozdev V.E., Blinova D.V., Rovneiko N.I., Yamalova O.P.** Sistemnoe i strukturnoe modelirovanije trebovanij k programmnhim produktam i proektam [System and structural modelling of software projects and products requirements] // Programmnaia injenerija. Teoreticheskij i prikladnoi nauchno-tehnicheskij zhurnal, №5, 2013 – pp. 2-10 (In Russian).

- [22] ESA PSS-05-0 European Space Agency software engineering standards, February 1991  
URL: <ftp://ftp.estec.esa.nl/pub/wm/anonymous/wme/bssc/PSS050.pdf> (Valid on 05.10.2015)
- [23] ISO/IEC 15288:2002 «System engineering — System life cycle processes»
- [24] **Kaplan R.S., Norton D.P.** The Balanced Scorecard: Translating Strategy into Action. – Harvard Business Review Press, 1996. – 336 p.
- [25] **Gvozdev V.E., Subhangulova A.S., Bezhaeva O.Ya.** Otsenka kolichestva defektov v programmnykh modulakh na osnove pokazatelei slozhnosti [Estimate of the numbers of defects in program modules based on index of difficulty] // Informacionnie tehnologii i sistemi: proc. of 4<sup>th</sup> Int. conf., Bannoe, Russia, 2015. – pp. 12-14 (In Russian).
- [26] **Lyu M.R.** Handbook of Software Reliability Engineering, IEEE and McGraw-Hill, 1996. – 850p.
- [27] **Lee D.Y., Vouk M., Williams L.** Using Software Reliability Models for Security Assessment - Verification of Assumptions, // ISSRE 2013, Pasadena, 4-7 Nov 2013, pp 23-24
- [28] IEC 61165:1995 Application of Markov techniques
- [29] CMMI for Development, Version 1.3, Software Engineering Institute, 2010.  
URL: <http://www.sei.cmu.edu/reports/10tr033.pdf> (Valid on 24.06.2015).
- [30] IEEE Guide to the Software Engineering Body of Knowledge. SWEBOK, 2004.  
URL: <http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf> (Valid on 24.06.2015)
- [31] **Casti J.** Connectivity, complexity, and catastrophe in large-scale systems. New York University, International Institute for Applied Systems Analysis, JOHN WILEY & SONS Chichester – New York – Brisbane – Toronto, 1979. – 203 p.
- [32] **Gvozdev V.E., Guzairov M.B., Piyasov B.G., Bezhaeva O.Ya.** Elementi sistemnoi injenerii: metodologicheskije osnovi razrabotki programmnykh sistem na osnove V-modeli zhiznennogo tsikla [Elements of system engineering: methodological basis of program system development based on V-model of lifecycle]. – M.: Mashinostroeniye, 2013. – 180 p. (In Russian).
- [33] **Vittikh V.A.** Situational management from the position of postneoclassic science / Ontology of Designing, № 2 (4), 2012. – pp.7-15. (In Russian).
- [34] **Sergeeva L.A.** Aktualnost sistemnogo podhoda v sovremennom informatsionnom obshchestve [Actuality of system approach in modern information society] // Filosofskie problemi infirmatsionnykh tehnologiy i kiberprostranstva, № 2, 2011 pp. 248-254 (In Russian).

## Сведения об авторах



**Гвоздев Владимир Ефимович**, 1955 г. рождения. Окончил Уфимский авиационный институт в 1978 г., д.т.н. (2000), профессор. Заведующий кафедрой технической кибернетики Уфимского государственного авиационного технического университета. В списке научных трудов более 250 работ, в том числе монография и учебные пособия по проектированию и реализации программных продуктов и проектов. Проводит исследования в области открытых информационных систем, прикладной статистики, теории надёжности, контроля и управления состоянием окружающей среды, управления программными проектами, функциональной безопасности.

**Gvozdev Vladimir Efimovich** (b. 1955) graduated from the Ufa Aviation Institute in 1978, D. Sc. Eng. (2000), prof. He is Head of Technical Cybernetic Department at Ufa State Aviation

Technical University. He is co-author of more than 150 publications in the field of design and production of program products and projects, open information systems, applied statistic, theory of reliability, monitoring and management of environmental condition, function safety.



**Блинова Дарья Викторовна**, 1984 г. рождения. Окончила Уфимский государственный авиационный технический университет в 2006 г., к.т.н. (2009). Доцент кафедры технической кибернетики Уфимского государственного авиационного технического университета. Проводит исследования в области прикладной статистики, управления природно-техническими системами, функциональной безопасности, дефектологии компонентов аппаратно-программных комплексов.

**Blinova Darya Viktorovna** (b.1984) graduated from the Ufa State Aviation Technical University in 2006, PhD (2010). She is Assistant Professor at Ufa State Aviation Technical University (Department of Technical Cybernetic). She conducts research in the field of applied statistic, management of natural and technical systems, function safety, defectology of hardware-software complexes components.