Системные архетипы как методическая основа обеспечения функциональной безопасности аппаратно-программных комплексов

О.Я. Бежаева

***Аннотация* — В настоящей статье рассматриваются методические основы обеспечения функциональной безопасности аппаратно-программных комплексов (АПК). Концептуальную основу исследований составляет системное сочетание проактивного, активного и реактивного подходов к управлению дефектами разной природы. Рассмотрение дефектов как разновидности сложных систем создает методологическую основу для научно-обоснованной адаптации подходов, методов и моделей, хорошо зарекомендовавших себя при решении задач управления сложными системами иной природы, в область управления функциональной безопасностью АПК. Системные архетипы являются концентрированной формой представления проблемных ситуаций, встречающихся при управлении сложными системами разной природы. В статье рассмотрено содержание системных архетипов применительно к проблеме обеспечения функциональной безопасности АПК. Предложены модели системных архетипов применительно к проблеме обеспечения функциональной безопасности. Предложенный подход положен в основу для представления когнитивных моделей проблемной ситуации посредством системных архетипов.**

***Ключевые слова* — субъектоцентрические системы, дефекты, функциональная безопасность, системные архетипы.**

I. Введение

В литературных источниках, посвященных проблематике реализации программных проектов, многократно подчеркивается такая особенность проектов как «уникальность». Меньшее внимание уделяется тому, что «роднит» программные системы со сложными субъектоцентрическими системами иной природы. Одной из характеристик такого «родства» является то общее, что именуется системными архетипами. Системные архетипы являются концентрированной формой представления подобных ситуаций, встречающихся при управлении сложными системами разной природы.

Дефекты – неотъемлемая составляющая субъектоцентрических систем. Это утверждение обосновывается в работах [1, 2]. В силу того, что аппаратно-программные комплексы (АПК) являются разновидностью субъектоцентрических систем, можно утверждать, что наличие латентных дефектов как в аппаратной, так и в программной компонентах, является неотъемлемой особенностью АПК.

Примером может служить известное высказывание Ф. Брукса «No Silver Bullet». Брукс утверждает, что «…ни в одной технологии или в управленческой технике не существует универсального метода, увеличивающего на порядок производительность, надёжность и простоту…» [2]. Иными словами, несмотря на постоянное развитие технологий, делающих возможным создание в условиях ограниченных ресурсов все более сложных аппаратно-программных комплексов, невозможно создать методологию, гарантирующую отсутствие латентных дефектов разной природы как в аппаратной, так и в программной составляющих этих субъектоцентрических систем.

Разные дефекты имеют различную природу возникновения. Характер дефектов зависит от того, на какой стадии жизненного цикла объекта они возникли. Тяжесть последствий от выявления дефекта зависит от интервала времени между моментом возникновения дефекта и его выявлением. В работе [3] отмечается, что чем больше интервал времени между возникновением и выявлением дефекта, тем больше изменений приходится вносить в инструкцию программного продукта. «Спусковым крючком» проявления одного и того же дефекта могут служить инициирующие события разной природы [4, 5].

Для модельного описания дефектов могут использоваться различные формальные подходы (алгоритмические и структурные модели) [5 - 7], математические модели [8, 9], т.е. дефекты как объекты моделирования обладают свойством полиморфизма. Отмеченные свойства не дают исчерпывающего описания особенностей дефектов.

Неопределенность является фактором риска программных проектов. Неопределенность, в том числе,

Статья получена 27 января 2021.
Работа поддержана грантом Российского Фонда Фундаментальных Исследований №19-08-00177 «Методологические, теоретические и модельные основы управления функциональной безопасностью аппаратно-программных комплексов в составе распределенных сложных технических систем»

О. Я. Бежаева, Уфимский государственный авиационный технический университет, Уфа, Россия (e-mail: obezhaeva@gmail.com)

обусловлена тем, что сложные системы фактически не имеют границ [10].

Границы систем – это мысленные модели, создающие предпосылки для построения моделей, характеризующих структуру (устройство) и поведение (внешнее реакцию системы на воздействия). Условность границ системы подводит к заключению, что совершенствование методов и технологий обеспечения функциональной безопасности не может гарантировать полного отсутствия латентных дефектов в силу того, что объектом управления является некая модель реальной системы, обозначенная границами системы. Как только мы пытаемся раздвинуть границы системы, возникают новые источники дефектов. А так как любая открытая система безгранична, число источников дефектов также безгранично. Тем не менее позволяют сделать заключение о том, что дефекты АПК следует рассматривать как разновидность сложных систем.

II. Основа обеспечения функциональной безопасности АПК

Архетип –в переводе с греческого «начальный образ», формальный прообраз, идея. В литературных источник [10 - 12] определяется роль архетипов как инструмента структуризации схожих по содержанию проблем, возникающих при управлении сложными системами. Отмечается, что «…Архетипы в менеджменте и управлении – грань, которая отделяет порядок от хаоса. Построение архетипа - предвестник успеха любого дела…».

Следует подчеркнуть связь между опытом, аксиологическими знаниями субъектов, участвовавших в реализации проектов создания компонентов информационно-вычислительных систем, и архетипным представлением проблемных ситуаций. В процессе развития компетенций разработчиков у них накапливается опыт решения задач, связанных с реализацией проектов, результатом чего является формирование описания проблемных ситуаций посредством архетипов. Следует подчеркнуть, что ориентация на опыт подчеркивает то обстоятельство, что речь идет об управлении сложной субъектоцентрической системой в условиях неопределенности, когда принять рациональной решение на основе подходов, ориентированных на управление в условиях четких целей и ограничений на решения, не представляется возможным.

Описание проблемных ситуаций посредством системных архетипов способствуют её структуризации, что создает предпосылки для её адекватного восприятия и как следствие выработки рациональных подходов к её урегулированию. В [11] отмечается, что архетипное представление проблемных ситуаций создает условия для применения знаний иных дисциплин – психологии, когнитологии, математики, социологии – для выявления возможных сценариев развития ситуаций, отражаемых архетипами в тех или иных обстоятельствах. Полученные при исследовании структурированных проблемных ситуаций результаты создают основу для построения в различной степени формализованных знаковых моделей (от когнитивных до математических). Наличие таких моделей позволяет оценить последствия различных организационных решений, иными словами, создают основу для предупреждения возникновения латентных дефектов организационной природы, влекущих за собой наиболее тяжелые негативные последствия [1].

Системные архетипы являются концентрированной формой представления проблемных ситуаций, встречающихся при управлении сложными системами разной природы.

Иными словами, архетипное представление проблемных ситуаций на разных стадиях жизненного цикла программных систем может способствовать выработке вариантов устойчивых организационных структур, соответствующих фактическому сочетанию внешней и внутренней сред проекта. Кроме того, реализовать проактивный подход к управлению программными проектами, основанный на прогнозных оценках: сценариях реализации этапов проектов; характеристиках качества предполагаемых результатов; требуемых ресурсах.

III. Содержание архетипов применительно к проблеме обеспечения функциональной безопасности АПК

Рассмотрим содержание чаще всего упоминаемых в литературе архетипов применительно к проблеме обеспечения функциональной безопасности АПК.

Архетип 1. Уравновешивание с задержкой.

Данному архетипу соответствует известная графическая модель, представленная на рисунке 1.

Рис.1 Графическая модель, соответствующая Архетипу 1

Примером этого архетипа применительно к проблеме обеспечения функциональной безопасности могут служить выявляемые на стадиях внедрения и эксплуатации несоответствия потребительских свойств потребностям системы управления (действительное состояние). Например, наличие у программных систем невостребованных функциональных возможностей [14]; неудовлетворительные характеристики надежности программных продуктов и т.д.

Причинами этого являются ошибки разной природы, следствиями которых являются организационные, конструкторские, технологические дефекты. Примером организационного дефекта может служить то, что при проведении предпроектного обследования игнорировались различия в потребностях и желаниях разных целевых групп пользователей. Примером

конструкторского дефекта может служить несоответствие в требованиях к уровню надежности программного продукта с точки зрения поддержки управления реализуемой им функции.

Установление причин возникновения дефектов и выработка мер по их устранению (корректирующие воздействия) требуют затрат времени (задержка). После этого выполняется новое сопоставление потребительских свойств системы и информационных потребностей системы управления. Вполне возможно, что в силу разных объективных и субъективных причин, за время выполнения корректировки потребительских свойств информационной системы изменятся потребности системы управления. Вновь запускается процесс внесения изменений в конструкцию информационной системы, т.е. возникает колебательный процесс.

Архетип 2. Пределы роста.

Данному архетипу соответствует известная графическая модель, представленная на рисунке 2.

Рис.2 Графическая модель, соответствующая Архетипу 2

Примером этого архетипа применительно к задачам обеспечения функциональной безопасности может служить, с одной стороны, стремление развивать и совершенствовать потребительские свойства систем информационной поддержки управления сложной системой в темпе изменения свойств как объекта управления, так и окружающей его среды (развивающий цикл, действия ведущие к улучшению). С другой стороны, создание новых компонентов инфокоммуникационных систем ведет к увеличению масштаба и сложности (иными словами неопределенности состояния систем информационной поддержки), что является причиной совершения разработчиками ошибок и возникновения латентных дефектов. Наличие дефектов ухудшает потребительские свойства систем информационной поддержки, т.е. дефекты являются фактором, препятствующим улучшению потребительских свойств информационной системы (стабилизирующий цикл, условия, препятствующие улучшению).

Создание новых технологий проектирования и реализации компонентов инфокоммуникационных систем играет роль действий, ведущих к улучшению. С другой стороны, невозможность выявить, формализовать и, соответственно, покрыть технологиями и инструментальными средствами все задачи, которые возникают при создании систем информационной поддержки, является фактором, препятствующим развитию потенциальности информационных технологий.

Архетип 3. Подмена проблемы.

Возникнувшая проблема может быть решена симптоматическим методом, однако в последующем эта проблема будет возникать вновь и вновь. Данному архетипу соответствует графическая модель, представленная на рисунке 3.

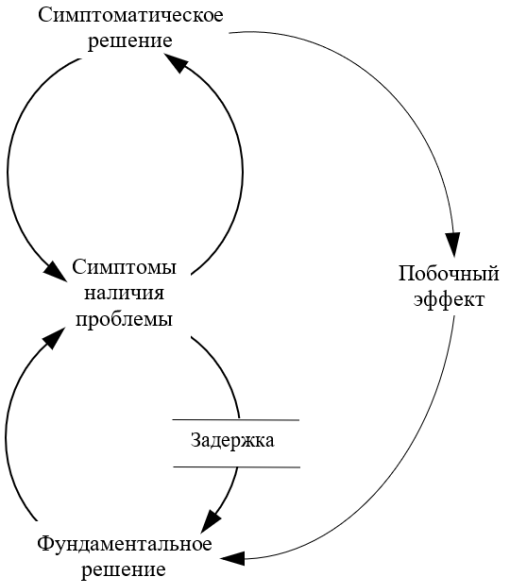

Рис.3 Графическая модель, соответствующая Архетипу 3

Примером реализации этого архетипа применительно к проблеме обеспечения функциональной безопасности служит деятельность по улучшению потребительских свойств компонентов инфокоммуникационных систем, основанная на локализации и устранении дефектов – непосредственных причин нежелательных событий (симптоматические решения). Основанием для поиска дефектов служит отклонение поведения компоненты инфокоммуникационной системы от желаемого с точки зрения пользователя (симптомы наличия проблемы). Однако устранение лиш ь непосредственных причин приводит к тому, что однотипные дефекты проявляются вновь и вновь. Это вынуждает выявлять коренные причины возникновения дефектов (например, посредством известного метода Root Cause Analysis – RCA) и вносить изменения в процессы реализации стадий жизненного цикла аппаратно-программных комплексов (фундаментальные решения). Выявление коренных причин требует затрат ресурсов, в том числе времени, что является задержкой улучшения потребительских свойств систем информационной поддержки.

Ограничением подхода к обеспечению функциональной безопасности, основанного на фиксации симптомов нежелательных событий, является его реактивная природа, т.е. невозможность за счет фундаментальных решений предупредить возникновение коренных дефектов иной природы, кроме тех, которые установлены посредством RCA. Такие коренные дефекты являются источником вторичных дефектов, которые фактически латентно

присутствуют в системе, но еще не проявлялись в процессе экстлуатации (побочный негативный эффект подхода).

Архетип 4. Размывание целей.

Этому архетипу соответствует графическая модель, представленная на рисунке 4.

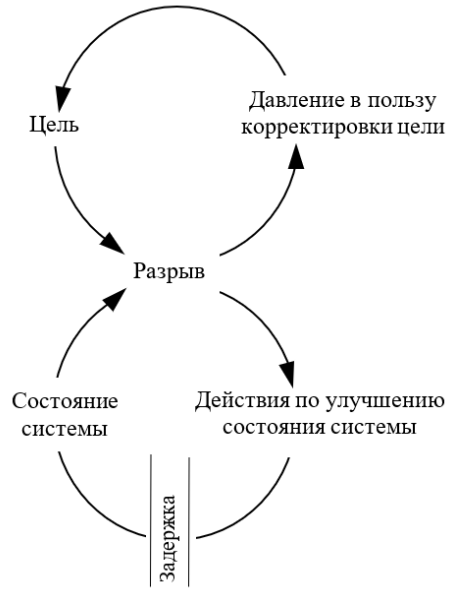

Рис.4 Графическая модель, соответствующая Архетипу 4

Примером этого архетипа применительно к проблеме функциональной безопасности может служить следующий. Представители различных целевых групп высказывают различные требования к организации пользовательского интерфейса, причем часть пожеланий ориентирована на использование устаревших технологий (например, использованию «командной строки» вместо GUI). Видя такой «разнобой» в требованиях (разрыв), разработчик фиксирует в задании требования к пользовательскому интерфейсу (цель) исходя из собственных субъективных предпочтений. При этом он рассчитывает на то, что после развертывания системы у пользователя и проведения обучения, потребители признают предлагаемое решение удачным (действия по улучшению состояния системы). На выполнение проектных и конструкторских работ затрачено время (задержка) и получен программный продукт с хорошим, с точки зрения разработчика, пользовательским интерфейсом (состояние системы). Продукт предъявлен представителям заказчика, пользовательский интерфейс получает низкую оценку различных целевых групп пользователей, т.к. вынуждает их изменять сложившиеся у них стереотипы взаимодействия с информационными системами. Разработчик, в соответствие с условиями договора (давление в пользу корректировки цели), в течение ограниченного срока обязан устранить замечания, что вынуждает его изменить цель реализации интерфейса и выполнить перепрограммирование.

Архетип 5. Эскалация (расширение).

Этому архетипу соответствует графическая модель, представленная на рисунке 5.

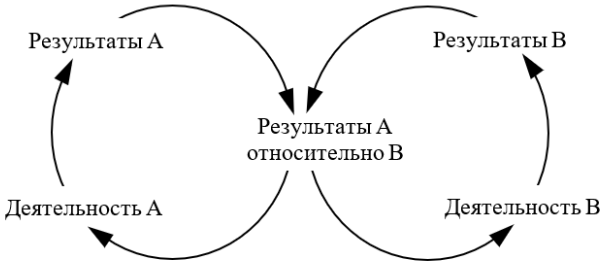

Рис.5 Графическая модель, соответствующая Архетипу 5

Примером этого архетипа применительно к обеспечению функциональной безопасности является ситуация, когда внедрение информационных технологий в организации, с одной стороны, приводит к расширению сфер влияния одних сотрудников за счет повышения эффективности решения бизнес-задач (контур Деятельность А - Результат А). С другой стороны, в силу перераспределения должностных полномочий приводит к сокращению у ряда сотрудников сфер влияния (контур Деятельность В – Результат В). Реакцией на складывающуюся ситуацию со стороны В является стремление закупить, освоить и внедрить в деятельность организации еще более современные информационные технологии, возможности которых превышают информационные потребности бизнес-процессов организации. Следствием такой «IT-гонки» является необоснованная трата ресурсов организации.

Архетип 6. Деньги к деньгам.

Этому архетипу соответствует следующая графическая модель, представленная на рисунке 6.

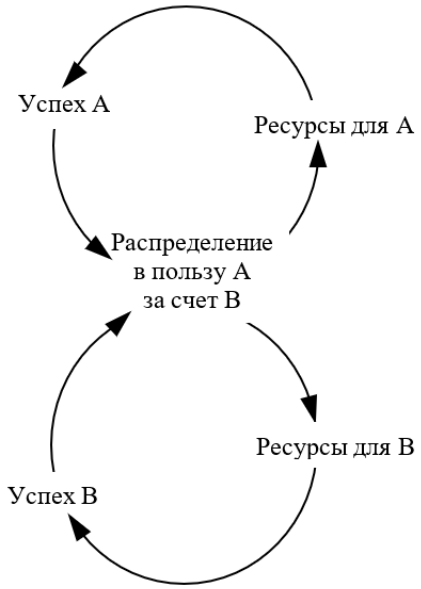

Рис.6 Графическая модель, соответствующая Архетипу 6

Примером этого архетипа применительно к обеспечению функциональной безопасности является ситуация, когда повышение качества спецификаций требований (пользователей; системных;

функциональных компонентов; модулей и т.д.) приводит к сокращению количества дефектов в конечных продуктах. Уменьшение количества дефектов сокращает затраты на их устранение (контур Ресурсы для В – Успех В), что делает возможным направить освободившиеся ресурсы (Распределение в пользу А за счет В) на развитие методической, инструментальной и образовательной базы обеспечения качества аппаратно-программных комплексов (Ресурсы для А – Успех А).

Архетип 7. Трагедия общих ресурсов.

Данному архетипу соответствует следующая графическая модель, представленная на рисунке 7.

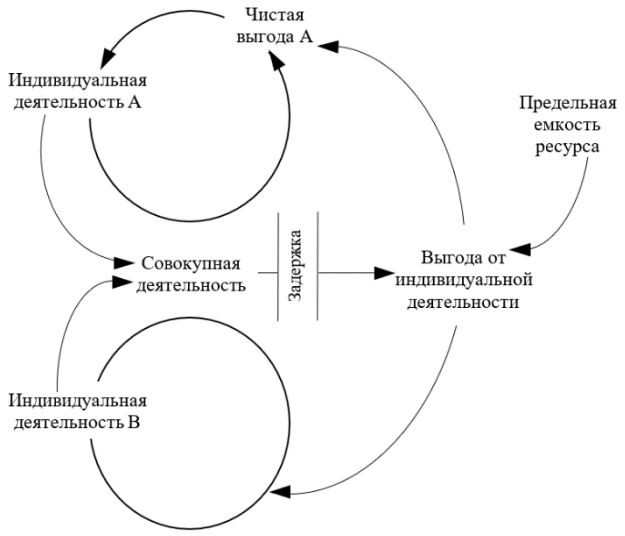

Рис.7 Графическая модель, соответствующая Архетипу 7

Примером реализации этого архетипа применительно к обеспечению функциональной безопасности является следующее. В условиях ограниченности бюджета и длительности реализации проекта руководством организации – Исполнителя на создание продукта выделяется больше ресурсов (контур Индивидуальная деятельность В), чем на комплексное тестирование (Индивидуальная деятельность А). Основанием для такого решения является известный тезис о том, что тестирование является затратным мероприятием и не повышает качества программного продукта [13]. Из-за недостатка ресурсов на проведение комплексных испытаний, пользователи получают продукт, содержащий много латентных дефектов (Совокупная деятельность). Систематически низкое качество поставляемых продуктов по прошествии некоторого времени (задержка) приводит к сокращению числа и объемов заказов (предельная емкость ресурса), т.е. сокращению вознаграждения каждого из сотрудников организации – Исполнителя (выгода от индивидуальной деятельности). Если не разорвать этот порочный круг, то все завершится ликвидацией организации – Исполнителя и профессиональными репутационными потерями её сотрудников.

Архетип 8. Неработающее решение. Данному архетипу соответствует следующая графическая модель, представленная на рисунке 8.

Рис.8 Графическая модель, соответствующая Архетипу 8

Примером реализации этого архетипа применительно к обеспечению функциональной безопасности является Стремление в условиях ограниченного срока реализации проекта (проблема), как можно раньше начать кодирование программного продукта (решение) без выполнения полноценного предпроектного обследования. Результатом такой стратегии являются многократные переделки разработок, напряженные межличностные отношения исполнителей (непредвиденные последствия) и, как следствие, превышение бюджета проекта, нарушение ограничений на длительность реализации проекта, в крайнем случае - провал проекта (задержка), т.е. неполучение заказчиком в нужное время необходимых для реализации его бизнес-процессов информационных ресурсов. [3, 14].

Архетип 9. Рост и недоинвестирование. Данному архетипу соответствует следующая графическая модель, представленная на рисунке 9.

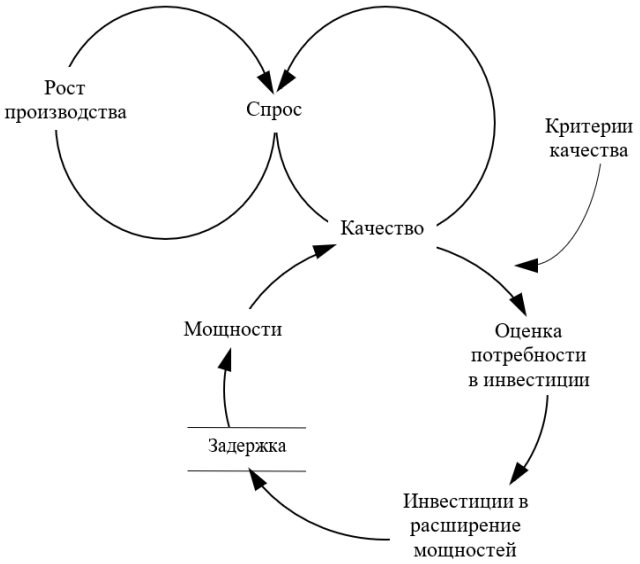

Рис.9 Графическая модель, соответствующая Архетипу 9

Примером этого архетипа может служить следующая ситуация. В условиях реализации положений доктрины Industry 4.0 и возрастания спроса на развитие «умного производства» возрастает спрос на создание цифровой экосреды предприятий, в том числе за счет интеграции

локальных информационных систем (контур «Рост производства – спрос»). При этом обязательным условием со стороны Заказчика выступает высокий уровень информационной и функциональной безопасности цифровой экосреды (контур «Спрос-качество»), а также жесткие ограничения на сроки завершения работ. Исходя из требований к уровню безопасности и ограничений на длительность реализации портфеля создания цифровой экосреды (критерии качества) были оценены потребности в инвестициях.

IV. Заключение

Неопределенность является неотъемлемым свойством программных проектов. Это обстоятельство фиксируется в известной модели «конус неопределенности». Неопределенность является фактором возникновения проблемных ситуаций на разных стадиях жизненного цикла программных систем, что, в свою очередь, может служить причиной совершения ошибок разной природы (организационных, проектных, технологических).

Таким образом, архетипное представление проблемных ситуаций, связанных с обеспечением функциональной безопасности аппаратно-программных комплексов, является предпосылкой научной адаптации эффективных подходов к управлению субъектоцентрическими системами разной природы. Иными словами, архетипное представление проблемных ситуаций на разных стадиях жизненного цикла программных систем может способствовать выработке вариантов устойчивых организационных структур, соответствующих фактическому сочетанию внешней и внутренней сред проекта.

Библиография

[1] J. Reason, E. Hollnagel, J. Paries, “Revisiting the “Swiss Cheese” Model of Accidents”, EEC Note No. 13/06. European Organization for the Safety of Air Navigation, October 2006, 25 p.

[2] Brooks, Frederick P., “No Silver Bullet: Essence and Accidents of Software Engineering”. Computer, Vol. 20, No. 4 (April 1987) pp. 10-19. (DOI: 10.1109/MC.1987.1663532)

[3] Скотт Беркун. “Искусство управления IT-проектами”. Издательство: Питер, 2007 г. 400с.

[4] G. Klein, D. Snowden, L.P. Chew, “Anticipatory Thinking” in *Proc. International NDM Conf.* (Eds. K. Mosier & U. Fischer), Pacific Grove, CA, June 2007, pp. 1-7 .

[5] R. Silva, M. Carvalho, “Anticipatory Failure Determination (AFD) for product reliability analysis: A comparison between AFD and Failure Mode and Effects Analysis (FMEA) for identifying potential failure modes” in *Federal Technological University of Paraná (UTFPR)*, Curitiba, Brazil, January 2019, 24p. doi: 10.1007/978-3-319-78075-7_12

[6] Гвоздев В.Е., Бежаева О.Я., Ахметова Д.Р. “Построение модели многосвязного объекта на основе совместного использования данных и экспертных оценок”. Онтология проектирования. – 2019. – Т.9, №3 (33). – С. 361-368.

[7] Гвоздев В.Е., Бежаева О.Я., Насырова Р.А. “Модели возникновения ошибок на предпроектной стадии разработки компонент информационно-вычислительных систем”. Онтология проектирования. – 2020. – Т.10, №1 (35). – С. 73-86.

[8] Липаев В.В. “Надежность программных средств”. М: Синтег, 1998, 232с.

[9] Липаев В.В. “Надежность и функциональная безопасность комплексов программ реального времени”. М: Институт системного программирования Российской академии наук. 2013, 176с.

[10] Donella H. Meadows. “Thinking in Systems: A Primer”. Chelsea Green Publishing, 2008, 240 p.

[11] Райков, А.Н. “Конвергентное управление и поддержка решений”. М.: Издательство ИКАР, 2009. - 245 с.

[12] Сенге Питер. “Пятая дисциплина”. М.: Олимп-Бизнес, 2003. 408 с.

[13] Майерс Г.Дж. Надежность программного обеспечения. М: Издательство Мир, 1980. – 359 с.

[14] CHAOS Report. The Standish Group International, Inc., 2018, 68 p. - Available: https://www.standishgroup.com/news/37

System archetypes as a methodological basis for ensuring the functional safety of hardware and software systems

O.Ya. Bezhaeva

***Abstract* - This paper discusses the methodological foundations of ensuring the functional safety of hardware and software complexes. The conceptual basis of the research is a systematic combination of proactive, active and reactive approaches to the management of different nature defects. The consideration of defects as a variety of complex systems creates a methodological basis for the scientifically based adaptation of approaches, methods and models that have proven themselves in solving problems of complex systems management of a different nature in the field of functional safety management. System archetypes are a concentrated form of representation of problem situations encountered in the management of complex systems of different nature. The description of problem situations through system archetypes contributes to its structuring, creates prerequisites for its adequate perception and, as a result, the development of rational approaches to its resolution. The results obtained in the study of structured problem situations form is the basis for the construction of various degrees of formalized sign models (from cognitive to mathematical). The presence of such models allows us to assess the consequences of various organizational decisions, in other words, create a basis for preventing the occurrence of latent defects of organizational nature, which entail the most negative consequences. The paper considers the content of system archetypes in relation to the problem of ensuring of the functional safety of hardware and software systems. Models of system archetypes are proposed in relation to the problem of ensuring functional safety. The proposed approach is the basis for the representation of cognitive models of a problem situation by means of system archetypes.**

***Keywords* — subject-centric systems, defects, functional safety, system archetypes.**

REFERENCES

[1] J. Reason, E. Hollnagel, J. Paries, “Revisiting the “Swiss Cheese” Model of Accidents”, EEC Note No. 13/06. European Organization for the Safety of Air Navigation, October 2006, 25 p.

[2] Brooks, Frederick P., “No Silver Bullet: Essence and Accidents of Software Engineering”. Computer, Vol. 20, No. 4 (April 1987) pp. 10-19. (DOI: 10.1109/MC.1987.1663532)

[3] Skott Berkun. “Iskusstvo upravlenija IT-proektami”. Izdatel'stvo: Piter, 2007 g. 400c.

[4] G. Klein, D. Snowden, L.P. Chew, “Anticipatory Thinking” in *Proc. International NDM Conf.* (Eds. K. Mosier & U. Fischer), Pacific Grove, CA, June 2007, pp. 1-7 .

[5] R. Silva, M. Carvalho, “Anticipatory Failure Determination (AFD) for product reliability analysis: A comparison between AFD and Failure Mode and Effects Analysis (FMEA) for identifying potential failure modes” in *Federal Technological University of Paraná (UTFPR)*, Curitiba, Brazil, January 2019, 24p. doi: 10.1007/978-3-319-78075-7_12

[6] Gvozdev V., Munasipov R., Bezhaeva O., Akhmetova D. "Construction of a multi-connected object model based on the joint use of data and expert evaluation". Design ontology. - 2019. - Vol. 9, No. 3 (33). - p. 361-368

[7] Gvozdev V. E., Bezhaeva O. Ya., Nasyrova R. A. "Models of errors at the pre-design stage of the development of information and computing systems components". Design ontology. - 2020. - Vol. 10, No. 1 (35). - pp. 73-86.

[8] Lipaev V. V. "Reliability of software tools". M: Sinteg, 1998, 232s.

[9] Lipaev V. V. "Reliability and functional safety of real-time program complexes". M: Institute of System Programming of the Russian Academy of Sciences. 2013, 176c.

[10] Donella H. Meadows. “Thinking in Systems: A Primer”. Chelsea Green Publishing, 2008, 240 p.

[11] Raikov, A. N. "Convergent management and decision support". Moscow: IKAR Publishing House, 2009. - 245 p.

[12] Peter Senge. "The fifth discipline". Moscow: Olymp-Business, 2003. 408 p.

[13] Myers G. J. Software reliability. Moscow: Mir Publishing House, 1980. - 359 p.

[14] CHAOS Report. The Standish Group International, Inc., 2018, 68 p. - Available: https://www.standishgroup.com/news/37