## Технологическая и эксплуатационная безопасность программ.

Необходимость определения этапов жизненного цикла (ЖЦ) ПО обусловлена стремлением разработчиков к повышению качества ПО за счет оптимального управления разработкой и использования разнообразных механизмов контроля качества на каждом этапе, начиная от постановки задачи и заканчивая авторским сопровождением ПО. Наиболее общим представлением жизненного цикла ПО является модель в виде базовых этапов - процессов, к которым относятся:

* системный анализ и обоснование требований к ПО;
* предварительное (эскизное) и детальное (техническое) проектирование ПО;
* разработка программных компонент, их комплексирование и отладка ПО в целом;
* испытания, опытная эксплуатация и тиражирование ПО;
* регулярная эксплуатация ПО, поддержка эксплуатации и анализ результатов;
* сопровождение ПО, его модификация и совершенствование, создание новых версий.

Данная модель является общепринятой и соответствует как отечественным нормативным документам в области разработки программного обеспечения, так и зарубежным. С точки зрения обеспечения технологической безопасности целесообразно рассмотреть более подробно особенности представления этапов ЖЦ в зарубежных моделях, так как именно зарубежные программные средства являются наиболее вероятным носителем программных дефектов диверсионного типа.

Графическое представление моделей ЖЦ позволяет наглядно выделить их особенности и некоторые свойства процессов.

Первоначально была создана каскадная модель ЖЦ, в которой крупные этапы начинались друг за другом с использованием результатов предыдущих работ. Наиболее специфической является спиралевидная модель ЖЦ. В этой модели внимание концентрируется на итерационном процессе начальных этапов проектирования. На этих этапах последовательно создаются концепции, спецификации требований, предварительный и детальный проект. На каждом витке уточняется содержание работ и концентрируется облик создаваемого ПО.

Стандартизация ЖЦ ПО проводится по трем направлениям. Первое направление организуется и стимулируется Международной организацией по стандартизации (ISO - International Standard Organization) и Международной комиссией по электротехнике (IEC - International Electro-technical Commission). На этом уровне осуществляется стандартизация наиболее общих технологических процессов, имеющих значение для международной кооперации. Второе направление активно развивается в США Институтом инженеров электротехники и радиоэлектроники (IEEE - Institute of Electrotechnical and Electronics Engineers) совместно с Американским национальным институтом стандартизации (American Na-tional Standards Institute-ANSI). Стандарты ISO/IEC и ANSI/IEEE в основном имеют рекомендательный характер. Третье направление стимулируется Министерством обороны США (Department of Defense-DOD). Стандарты DOD имеют обязательный характер для фирм, работающих по заказу Министерства обороны США.

Для проектирования ПО сложной системы, особенно системы реального времени, целесообразно использовать общесистемную модель ЖЦ, основанную на объединении всех известных работ в рамках рассмотренных базовых процессов. Эта модель предназначена для использования при планировании, составлении рабочих графиков, управлении различными программными проектами.

Совокупность этапов данной модели ЖЦ целесообразно делить на две части, существенно различающихся особенностями процессов, технико-экономическими характеристиками и влияющими на них факторами.

В первой части ЖЦ производится системный анализ, проектирование, разработка, тестирование и испытания ПО. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих этапах существенно зависят от объекта и среды разработки. Изучение подобных зависимостей для различных классов ПО позволяет прогнозировать состав и основные характеристики графиков работ для новых версий ПО.

Вторая часть ЖЦ, отражающая поддержку эксплуатации и сопровождения ПО, относительно слабо связана с характеристиками объекта и среды разработки. Номенклатура работ на этих этапах более стабильна, а их трудоемкость и длительность могут существенно изменяться, и зависят от массовости применения ПО. Для любой модели ЖЦ обеспечение высокого качества программных комплексов возможно лишь при использовании регламентированного технологического процесса на каждом из этих этапов. Такой процесс поддерживается CASE средствами автоматизации разработки, которые целесообразно выбирать из имеющихся или создавать с учетом объекта разработки и адекватного ему перечня работ.

<http://emag.iis.ru/arc/infosoc/emag.nsf/BPA/ca253fcdd6e78e15c325763f003ffe87>

**Технологическая безопасность програмного обеспечения - новая проблема в области создания информационных систем**

В последние годы отечественные и зарубежные ученые пришли к весьма важному в методологическом плане выводу, что проблема обеспечения безопасности информации в процессе Штатной эксплуатации автоматизированных систем постепенно отходит на второй план. Появление средств вирусного заражения программного обеспечения (ПО) несколько "подстегнуло" ученых, работающих в области практического программирования, что привело при рассмотрений вопросов защиты информации к некоторому смещению внимания на этап эксплуатации информационных и вычислительных систем. При этом проблемы, связанные с технологиями разработки ПО для систем критических приложений (оборонных, энергетических, транспортных, коммутационных, финансовых и экологически опасных производств), остались без должного внимания [1]. Данные проблемы являются новыми, недостаточно исследованными и почти не представленными в литературе.

Воздействие на информацию в процессе функционирования автоматизированных систем осуществляется в целях несанкционированного ее раскрытия, изменения или уничтожения. Решение задач, связанных с предотвращением воздействия непосредственно на информацию, осуществляется в рамках комплексной проблемы защиты информации и имеет достаточно развитую научно-методическую базу«[2]. Данный аспект обеспечения безопасности информации и средств ее обработки именуется эксплуатационной безопасностью, так как соответствует этапу применения автоматизированных систем.

Анализ, проведенный специалистами Министерства обороны, Гостехкомиссии России, Российской академии наук и промышленных организаций, показал, что функциональные и надежностные характеристики систем критических приложений в значительной степени определяются качеством и надежностью ПО, входящего в их состав. В то же время технологии разработки ПО, составляющие основу современных' информационных технологий, в настоящее время далеко не совершенны. Недостатки этих технологий приводят к тому, что ПО не является защищенным от ошибок людей и отказобезопасным, а его применение в составе автоматизированных систем приводит к риску и неопределенности, которые должны быть учтены при принятии решения о практическом использовании этих систем.

Кроме проблем качества и надежности ПО при создании систем критических приложений все большую актуальность приобретает фундаментальная проблема его безопасности. Данная проблема с точки зрения информационных технологий разработки ПО. является новой и связана с возможностью внедрения в тело программных средств на этапе их разработки (или модификации в ходе авторского сопровождения),, так называемых программных закладок.

Теоретически "программные закладки" рассматриваются как активное средство для целенаправленного воздействия на функционирование системы, в составе которой используется вычислительная техника. В связи с этим "программные закладки" рассматриваются как одна из разновидностей преднамеренных дефектов, а проблема защиты от внедрения таких дефектов выносится в самостоятельный раздел фундаментальной проблемы безопасности ПО, который называется технологической безопасностью программ.

Экспертами принято выделять автоматические и управляемые "программные закладки". Автоматические "программные закладки", как правило, имеют механизм срабатывания, который заранее настроен на условия функционирования автоматизированной системы в целом. Управляемые "программные закладки", в отличие от автоматических, имеют контролируемый извне механизм их активизации.

Если рассматривать обобщенную модель жизненного цикла программ (рисунок), то можно заметить, что основные классы нарушения технологической безопасности ПО распределяются между этапами создания программ от системного анализа условий применения разрабатываемого ПО до проведения испытаний готовых программных комплексов. При этом к основным классам нарушения технологической безопасности относятся:

* преднамеренное искажение требований к характеристикам ПО;
* использование средств автоматизации программирования, закладывающих логические ошибки и дефектные блоки в машинные коды программ;
* преднамеренное искажение управляющих и информационных связей между компонентами программного комплекса;
* преднамеренное исключение из программы и методики испытаний проверок ветвей, содержащих дефекты;
* формирование внешних признаков нормального функционирования программ.

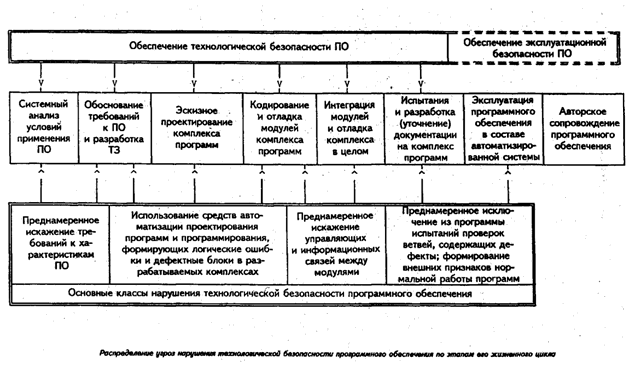
Анализ рассмотренных классов нарушений позволил сделать вывод, что основной целью обеспечения технологической безопасности программных средств является предотвращение возможности нарушения режимов функционирования компонентов ПО и автоматизированных систем путем исключения (существенного затруднения) внедрения в программы дефектов диверсионного типа. При этом достижение главной цели возможно только в том случае, если обеспечивается:

* предотвращение условий нарушения единой технологии разработки программных средств, охватывающей все этапы их жизненого цикла;
* контроль технологической безопасности инструментальных средств, применяемых для автоматизации программирования;
* локализация программных дефектов, выявленных в ходе тестирования (испытания) программных средств;
* ликвидация последствий воздействия на функционирование программ .обнаруженных и локализованных дефектов.

Однако в настоящее время отсутствует единая технология разработки программных средств систем критических приложений, инструментальные средства практически не контролируются, а готовые компоненты ПО не проверяются на отсутствие программных закладок, внесенных в них в процессе промышленного производства. Данные обстоятельства обусловлены в первую очередь отсутствием научно-методической базы обеспечения технологической безопасности программ.

За рубежом проблема технологической безопасности программного обеспечения, используемого в системах военного назначения, системах управления ядерными реакторами, экологически опасными производствами, финансовыми ресурсами и в других системах критических приложений, решается на государственном уровне: в США — в рамках программы STARS, во Франции — LTR-3, в Великобритании — ALVAY, в ЕЭС — ESPRIT. Кроме того, в США законодательно запрещено применение иностранных вычислительных средств и информационных технологий при разработке программных продуктов, используемых в системах критических приложений.

В Российской Федерации вопросы технологической безопасности программного обеспечения систем критических приложений решаются только в плане научных исследований при существующих организационно-технических и финансовых ограничениях. Это не позволяет устранить угрозу поражения ПО дефектами диверсионного типа, однако уже сегодня можно определить основные задачи, которые необходимо решить для повышения эффективности обеспечения технологической безопасности программных средств перспективных систем. К этим задачам относятся:



а) создание методологии и поддерживающего ее инструментария для интеграции, взаимообмена и накопления типовых компонентов ПО, полученного и проверенного в процессе проектирования, разработки, сопровождения надежных программных комплексов;

б) создание системы унификации и стандартизации для формального задания характеристик интегрируемых компонент, используя принципы архитектуры открытых систем (OSI). Распространение системы гибкой унификации не только и не столько на уровень программ и пакетов, сколько на уровень асинхронных или параллельных процессов и виртуальных машин;

в) создание системы поддержки всего цикла проектирования, разработки и сопровождения программ в виде CASE-системы, обеспечивающей:

инкриминтальность процесса разработки любого компонента программных средств на основе его сопровождения исчерпывающим комплектом документации с ранних этапов проектирования;

многовариантность проектирования, что позволяет комплексировать из хранящихся в фонде типовых компонентов конкретную версию программного комплекса с учетом специфики его применения;

управление качеством проектирования путем выбора проектных или реализационных версий на основе метрик оценки трудоемкости, ресурсоемкости, сложности проекта;

архивацию накопленных решений в сети проекта, обеспечивающих не только версионность, но и откат по проектным решениям к ранее принятым вариантам. Этот процесс должен обеспечивать межпроектный обмен компонентами программ на заданном уровне;

сборку экспериментального или реализационного варианта системы вместе с окружением для создания модели, прототипа, тестируемого макета, целевого программного комплекса с учетом развития в направлении автоматической генерации кода по спецификациям;

г) создание методологии, инженерных методик и моделей распределенного программного обеспечения в виде сетей взаимодействующих процессов и окружения, которые должны решать задачи:

поддержки ранних этапов анализа и проектирования спецификаций, оптимального выбора проектных решений на основе методик и метрик оценки и прогнозирования трудоемкости, ресурсоемкости и сложности вариантов;

контроля корректности реализации относительно спецификаций и верификации протоколов взаимодействующих процессов, выбора критериев оценки эффективности, сравнения и аттестации программных комплексов и их компонентов;

д) создание методов и средств контроля технологической безопасности программного обеспечения, используемых на этапах проектирования, разработки и испытания сложных комплексов программ.

Даже предварительный анализ перечисленных задач позволяет сделать вывод о том, что их решение потребует концентрации усилий большого количества специалистов в течение длительного времени. Однако уже сегодня можно утверждать, что наиболее приоритетной задачей является разработка научных основ исследования внутренних свойств дефектов диверсионного типа и моделей их проявления в составе ПО.

Теоретической основой исследования преднамеренных программных дефектов диверсионного типа является дефектология ПО, которая в рамках теоретического программирования изучает все свойства ПО, связанны о дефектами в программах и все свойства самих программных дефектов. Дефектология ПО базируется на фундаментальных свойствах самого ПО: полирутности ("ветвистости"), энтропийной сложности, идентичной копируемое и локальной стабильности. Основными дефектологическими свойствами ПО являются следующие: дефектоскопичность, дефектабельность, дефектогенность, дефектомобильность, дефектодиагностируемость, дефектокорректируемость, дефектоустойчивость.

Дефектоскопичность — это свойство ПО проявлять дефекты в некоторых условиях эксплуатации.

Дефектабельность — это имманентное свойство ПО содержать дефекты.

Дефектогенность — это свойство, характеризующее возможность поражения ПО дефектами (при этом, естественно, учитываются и технологическая, аппаратная и антропогенная среды).

Дефектомобильность, дефектодиагностируемость, дефектокорректируемость, дефектоустойчивость описывают возможности соответственно: передвижения (миграции) дефектов в ПО; обнаружения дефектов в ПО; устранения обнаруженных дефектов из ПО; противодействия (компенсации) негативных последствий дефектов, проявляющихся в процессе эксплуатации.

Для описания свойств самих дефектов в ПО служат такие дисциплины, как программная дефектоника и программная дефектометрия. Дефектоника и дефектометрия изучают структурные свойства программных дефектов и степень их негативного влияния на ход вычислительного процесса.

Разделом общей дефектологии ПО является криминальная дефектология, изучающая, в частности, такие свойства ПО, как диверсионная дефектогенноеть, активная дефектабельность, латентная дефектоскопия и управляемая дефектомобильность.

Дефектология ПО составляет основу нового научного направления — дефектологии компьютерной инфосферы (здесь должны быть добавлены исследования соответствующих системных связей между составляющими частями компьютерной инфосферы и интерфейса со всей инфосферой). В целом это научное направление сейчас находится в стадии становления, отдельные его разделы развиты неравномерно (наиболее широко представлена в научно-технической литературе дефектоскопия, являясь основной базой для современной теории надежности ПО).

Перспективным для развития дефектологии компьютерной инфосферы и ее разделов и решения на этой базе проблемы как технологической, так и эксплуатационной безопасности ПО может быть более глубокое использование результатов, полученных в рамках теоретического программирования. Так, для диверсионной дефектогенности весьма эффективным может стать применение подхода, изложенного в журнале "Компьютер Пресс" (1991 г., № 10), согласно которому всякая развитая программная система имеет склонность к расползанию, самодеструкции и, как первый признак, к неконтролируемой репликации своих частей. Фундаментальный математический результат, предсказывающий данное положение, был давно получен Дж. фон Нейманом в теореме, сущность которой состоит в том, что конечный автомат, достигший определенного уровня сложности, может реплицировать себя.

Создание научно-теоретического базиса, безусловно, совершенно необходимо для решения проблемы обеспечения технологической безопасности ПО систем критических приложений. Но кроме этого необходимы соответствующие организационные, административные и законодательные меры, причем безотлагательные, так как вое больше имеется сведений о проектах спецслужб развитых стран в области разработки средств (информационного оружия) поражения программных комплексов систем критических приложений.

Вероятность данной угрозы в современной обстановке резко возрастает из-за ряда факторов, основным из которых является широкое и бесконтрольное использование импортной вычислительной техники, программного обеспечения и информационных технологий при создании систем критических приложений- Дальнейшему распространению и использованию импортных информационных технологий способствует практически полное отсутствие их отечественных аналогов. В связи с этим под руководством Межведомственной комиссии по информационной безопасности при Совете Безопасности Российской Федерации разработана "Государственная программа создания безопасных информационных технологий", в рамках которой предполагается решить ряд первоочередных проблем, направленных на обеспечение технологической безопасности ПО систем критических приложений. Реализация данной программы позволит создать требуемую систему сбалансированных политических, юридических, организационных, научно-технических мероприятий как на государственном, так и на международном уровнях в интересах предотвращения несанкционированного воздействия на разрабатываемые системы.