

# Інформаційні технології

УДК 651. 34

Кассем Халифе, Т.Н. Шипова

Национальный технический университет «ХПИ», Харьков

## АНАЛИЗ И ИССЛЕДОВАНИЕ МОДЕЛЕЙ И МЕТОДОВ РАЗРАБОТКИ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*В статье проведены анализ и исследования моделей и методов разработки системного программного обеспечения компьютерных систем критического применения. Представлены факторы, влияющие на это процесс. Определены требования к качеству разработки системного программного обеспечения в целом и указано на приоритетность показателей безопасности программного обеспечения. Проведен анализ и сравнительное исследование основных подходов математического моделирования процесса разработки системного программного обеспечения. Определены приоритетные направления дальнейшего исследования.*

**Ключевые слова:** системное программное обеспечение, требования к качеству программного обеспечения, безопасность, SCRUM, Agile, математическое моделирование.

### Вступ

**Постановка проблемы.** В настоящее время Украинское законодательство понятие программного обеспечения определяет практически исключительно как объект авторского права и регламентирует процесс правового регулирования законом «Про авторське право і суміжні права» в редакции от 11 июня 2001 года. Однако ряд международных документов стандартизации программного обеспечения (ISO/IEC 25012:2008, IDT, ISO/IEC 25021:2012, IDT и др.) [8], подзаконных актов, принятых правительством Украины (ДСТУ ISO/IEC 25010:2016, ДСТУ ISO/IEC 25012:2016, ДСТУ ISO/IEC 25021:2016 и др.) [2] дают определение этой категории и регламентируют основные требования к качеству программного обеспечения (ПО). Анализ указанных документов, а также исследование специфики внедрения и функционирования ПО, позволил сделать вывод о значительном повышении требований к его качеству в условиях инсталляции и использования в компьютерных системах критического применения (КСКП).

Проведенный анализ требований показал, что более 54% требований к ПО КСКП связаны с обеспечением функциональной и информационной безопасности ПО. Следует заметить, что специфика ПО КСКП состоит в том, что более 70% его контента составляет системное ПО, обеспечивающее основные функции и жизнедеятельность систем критического применения. В то же время в силу своей приоритетности в процессе функционирования КСКП требования к безопасности системного ПО еще выше чем, у прикладного ПО.

В соответствии с руководящими документами [2, 8] в настоящее время существует ряд способов обеспечения безопасности системного ПО. Среди них выделим следующие:

- идентификация и аутентификация субъектов и объектов доступа;
- ограничение программной среды;
- защита машинных носителей информации;
- регистрация событий безопасности;
- антивирусная защита;
- обнаружение в системе управления компьютерных программ либо иной компьютерной информации, предназначенной для несанкционированного уничтожения, блокирования, модификации, копирования или нейтрализации средств защиты информации;
- контроль (анализом) защищенности информации;
- обеспечение целостности программной среды и информации;
- разработка «безопасных программ» или безопасное программирование;
- применение правовых методов защиты ПО и др.

Проведенные исследования показали, что вопросам обеспечения безопасности системного ПО уделяется достаточно большое внимание. Однако, в большей степени, используемые или предлагаемые методы защиты, касаются уже готовых продуктов, тем самым давая злоумышленникам возможности использовать «ошибки безопасности» системного ПО, допущенные и невыявленные разработчиками программного обеспечения. Поэтому, несмотря на многообразие методов и средств защиты системного ПО, вопросам разработки «безопасных программ» или безопасному программированию внимания уделяется недостаточно. В то же время, как показывает статистика киберпреступлений, именно этот недостаток становится причиной большинства успешно проведенных атак на КСКП.

Основные возможности устранения данного недостатка лежат в рамках управления разработкой су-

существующих методологий разработки ПО. Проведем анализ основных методологий разработки системного программного обеспечения и факторов, влияющих на этот процесс.

### 1. Анализ основных методологий разработки системного программного обеспечения и факторов, влияющих на этот процесс

Из литературы [1-8] известно, что системное ПО это совокупность программ для управления аппаратурой компьютера и обеспечения работы прикладных программ. Классификация системного ПО представлена на рис. 1.



Рис. 1. Классификация системного ПО

С точки зрения жизненного цикла программно-обеспечения и основных методологий разработки существенных различий в этапах разработки между прикладным и системным программным обеспечением нет. Однако, системное программное обеспечение разрабатывается так, чтобы компьютерная система могла эффективно выполнять свои функции. И именно от качества системного программного обеспечения зависят основные показатели функционирования КСКП. При разработке системного ПО, важным шагом является выбор методологии реализации этой задачи, позволяющей повысить эффективность и качество программного продукта. При этом необходимо учитывать, что совокупность процессов разработки системного ПО представляет собой сложную многофакторную систему прохождений этапов в рамках выбранной методологии.

Анализ литературы показал, что в основном современные методологии разработки ПО можно разделить на три основные группы: последовательные (прогнозируемые), циклические (полупрогнозируемые), гибкие (абстрактные). Следует заметить, что такое

разделение достаточно условно, так как в ходе выполнения поставленных задач возможны случаи изменения стратегий разработки. И тогда возникают ситуации пересечения данных методологий.

Проведенные исследования показали, что к числу последовательных методологий следует отнести каскадную модель (или модель водопада, Waterfall), которая характеризуется тем, что этапы разработки, такие как: анализ, проектирование, реализация, тестирование, идут друг за другом. Такая методика эффективно работает только тогда, когда требования стабильны и не меняются в ходе разработки, т.к. мы сразу описываем все требования, а потом сразу проектируем всю систему целиком [1].

К циклическим методологиям можно отнести V-Model и спиральную модель разработки. Имея ряд отличий друг от друга (например, в V-Model этап анализа и проектирования связан с этапом тестирования, а в спиральной модели на этапе анализа рисков создаются прототипы, концепты, модели которые призваны разрешить риск на ранней стадии) в целом данные модели обладают рядом достоинств, по сравнению с водопадной моделью. Это возможность незначительных отступлений в требованиях на повторных циклах, учет рисков в процессе разработки и т.д. Анализ деятельности известных фирм-разработчиков ПО показал, что в последнее время семейство гибких методологий занимает основное внимание на рынке. Основные принципы функционирования организаций в соответствии с данной категорией методологий сформулированы в документе, который получил название Agile Manifesto [5, 8]. В нем зафиксировано тринадцать основных принципов, среди которых выделим следующие (рис. 2): люди и их взаимодействие важнее процессов и инструментов; готовый продукт важнее документации по нему; сотрудничество с заказчиком важнее жестких контрактных ограничений; реакция на изменения важнее следования плану. Как видно из рис. 2 именно эти принципы характеризуют отличия между гибкими и последовательными методологиями и лежат в основе философии гибкой разработки ПО.

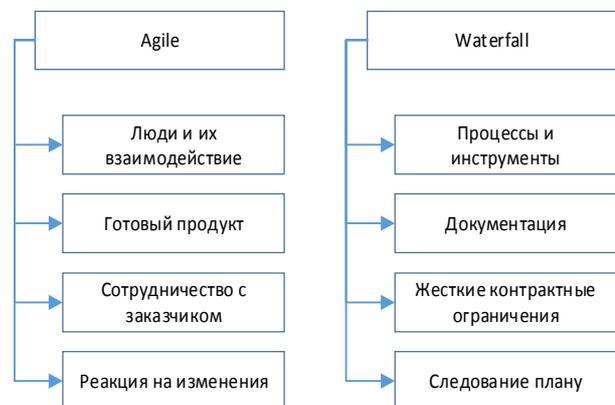


Рис. 2. Основные отличия между гибкими и последовательными методологиями

Проведенные исследования показали, что основным достоинством гибких методологий является ориентированность на проекты, где требования могут меняться по ходу разработки. В таких методологиях значительно ниже стоимость исправления ошибки.

В соответствии с результатами исследования Agile Survey [7] об использовании гибких методологий представим диаграмму их популярности в виде рис. 3.

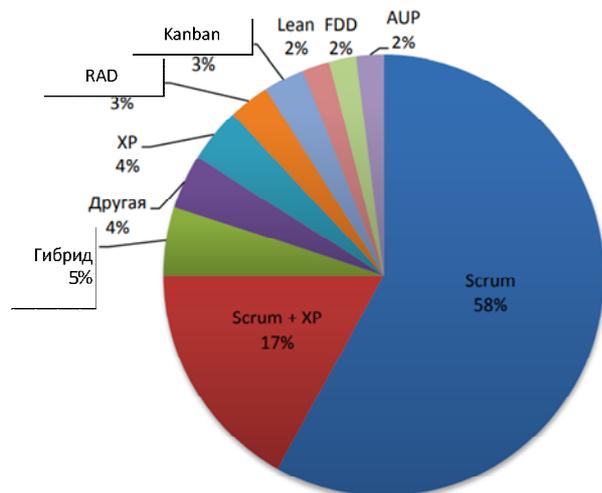


Рис. 3. Диаграмма популярности гибких методологий разработки ПО

Как видно из диаграммы среди гибких методологий можно выделить такие как: Scrum, XP, Kanban, RAD. При этом анализ деятельности разработчиков ПО в Украине показал, что сейчас наиболее популярной является методология Скрам (Scrum).

Проведенный анализ литературы показал, что Scrum — это каркас разработки, с использованием которого люди могут решать появляющиеся проблемы, при этом продуктивно и производя продукты высочайшей значимости. Авторами Scrum [7, 8] заявлены следующие особенности: легкий (англ. Lightweight); понятный, доступный.

Наряду с уже озвученными достоинствами, Scrum обладает и рядом недостатков. Среди них необходимо выделить следующие. Упор на самоорганизующуюся, многофункциональную команду. При кажущемся снижении затрат на координацию команды, это приводит к повышению затрат на отбор персонала, его мотивацию, обучение.

При определенных условиях рынка труда, формирование полноценной, эффективной Scrum команды может быть невозможным. Отсутствие плана коммуникаций и реагирования на риски [5] (в том числе и безопасности). Это в значительной степени сужает рамки использования методологии в проектах для КСКП. Анализ литературы показал, что методология экстремального программирования (XP) — состоит из 12 практик: парное программирование, разработка через тестирование, рефакторинг, простая архитектура, коллективное владение кодом, непрерывная инте-

грация, заказчик в команде, частые релизы, игра в планирование, 40-часовая рабочая неделя, стандарты кодирования, метафора системы [9].

Проведенные исследования показали, что также, как и Scrum, экстремальное программирование несет не только ряд преимуществ, но и содержит недостатки. В частности, это:

- сложности взаимодействия разработчиков при парном программировании;
- необходимость обеспечить постоянное присутствие заказчика в команде или заказчика в команде по вызову;
- сильная зависимость от устного общения;
- отсутствие стандартного способа описания дизайна и др.

Данные недостатки существенно затрудняют использование данной методологии для КСКП.

Анализ литературы показал, что методология Канбан (Kanban) представляет собой конвейер задач. Имеет всего 3 правила: визуализация процесса разработки с помощью канбан-доски, ограничение на количество задач на каждом этапе, постоянное измерение производительности команды и улучшения [4-6].

Основные преимущества системы Канбан:

- короткий производственный цикл, высокая оборачиваемость активов, в том числе запасов;
- отсутствуют или чрезвычайно низки издержки хранения производственных и товарных запасов;
- высокое качество продукции на всех стадиях производственного процесса и др.

Основные недостатки системы "точно в срок":

- сложность обеспечения высокой согласованности между стадиями производства продукции;
- значительный риск срыва производства и реализации продукции.

Следует заметить, что в последнее время все чаще разработчики ПО склоняются к комплексному использованию перечисленных выше методологий, что в значительной степени повышает эффективность работы коллектива.

Анализ литературы показал, что методология RAD (Rapid Application Development) — ориентирована на быструю разработку приложения, итеративно, с максимально простой архитектурой, минимальными издержками на процесс, максимально используя готовые компоненты и мощные инструменты [3].

Проведенные исследования показали, что на практике подобный подход к разработке приложений сопряжен с серьезными проблемами.

Все внимание сконцентрировано на экранных формах, а то, что касается правил обработки данных и системных функций, остается без внимания. Модули проектируются изолированно друг от друга. Следствием этого являются противоречия модулей, дублирование функций и данных, что может быть выявлено только при тестировании комплекса модулей. Функциональные возможности наращиваются параллельно

в нескольких направлениях, значит, структура базы данных должна контролироваться жестко.

Обработка исключительных ситуаций для каждого модуля производится своя [3].

Исходя из этого, RAD можно использовать лишь для небольших проектов, в которых определены требования бизнеса, ограничено количество интерфейсов и система ориентирована на экранные формы.

Таким образом, проведенные исследования показали, что несмотря на многообразие существующих методологий разработки ПО, все они не лишены ряда недостатков. В то же время их комбинации в процессе разработки может позволить повысить эффективность разработки и качество ПО.

Следует заметить, что в настоящее время не бывает универсального набора входных условий и внешних факторов воздействий для всех ситуаций при выборе той или иной методологии. В каждом случае необходимо рассматривать множество специфических факторов, которые должны быть ориентированы на специфику проекта. Особенно это важно в сложных проектах разработки системного ПО для КСКП. Требования таких систем обязывают разработчика уделять повышенное внимание вопросам безопасности ПО и учитывать факты повышения интереса злоумышленников именно к системному ПО. В работе за основу выберем методологию Scrum. Также следует отметить, что, выбрав эту методологию, необходимо ее адаптировать к повышенным требованиям качества (безопасности, надежности, производительности и др.).

## 2. Анализ и сравнительное исследование основных подходов математического моделирования процесса разработки системного программного обеспечения

Анализ литературы [5, 6, 10] показал, что в настоящее время существует множество подходов математического моделирования процесса разработки программного обеспечения в целом и системного ПО в частности. Их основой являются положения теорий компьютерной инженерии, массового обслуживания, нейронных сетей, нечеткой логики, графовых моделей и комбинаторных методов расчета и др. Кроме того, получили свое развитие средства решения оптимизационных задач, формализуемых на основе данных моделей. Это, прежде всего, аналитические методы, методы математического программирования, эвристические методы и др.. Проанализируем наиболее часто используемые на практике модели разработки системного ПО, которые в различной степени адаптированы к современным требованиям при формализации процессов управления разработкой.

Одним из современных направлений математического моделирования является биологическое направление с помощью нейронных сетей [10]. Во многом

это связано со спецификой функционирования компьютерных систем, которые являются человеко-машинными системами. Кроме того, в последнее время все большее внимание разработчики и проектировщики стали уделять вопросам защиты данных от программных угроз. А в этом случае результаты исследования систем, проведенные с помощью биологического подхода, показывают наиболее адекватные результаты [5-6].

Анализ систем биологического моделирования показал, что одним из наиболее популярных направлений в настоящее время является моделирование с помощью нейронных сетей. Связано это в основном с разработкой нейронной сети Хопфилда [10], в которой компьютерная сеть представляется в виде рекуррентной системы, решающей ряд специфических задач (прогнозирование, кластеризация, предварительного анализа и др.) При этом в качестве показателя качества системы выступает энергетическая функция вида:

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} w_{ij} x_i x_j, \quad (1)$$

где  $w_{ij}$  – весовой коэффициент между  $i$ -м и  $j$ -м нейронами;  $x_i$  и  $x_j$  – компоненты вектора  $X$  (входных данных системы), а в качестве критерия решения задачи распределения доступа и защиты данных в компьютерной сети используется выражение вида:

$$E = \min \left( -\frac{1}{2} \sum_{i=1}^N \sum_{k=1, k \neq i}^N \sum_{j=1}^{N_p(i)} \sum_{\ell=1}^{N_p(k)} |P_{ij} \cap P_{k\ell}| x_i^j x_k^\ell \right), \quad (2)$$

где  $P_{ij}$  –  $j$ -й ресурс между источником и адресатом;  $|P_{ij} \cap P_{k\ell}|$  – число узлов компьютерной сети, которые совместно используют маршруты  $P_{ij}$  и  $P_{k\ell}$ ;  $x_i^j = 1$ , если выбирается  $P_{ij}$ , и  $x_i^j = 0$  в обратном случае;  $N_p(i)$  – число вариантов направлений доступа, определенных между источником и адресатом.

Исследования генетического подхода моделирования показали, что он основан на идее, когда некий алгоритм может стать более эффективным, если позависствует лучшие характеристики у других алгоритмов («родителей»).

Анализ работ в области защиты данных [1-10] показал, что данное направление моделирования имеет ряд преимуществ, связанных с учетом специфики внешних воздействий на систему.

Однако, проведенные исследования моделей компьютерных систем, представленных в виде нейронных сетей [10] наряду с их достоинствами показали и недостатки связанные с существенными (до 100 наблюдений) временными затратами на процесс обучения при построении модели, и как следствие, «консерватизмом» по отношению к динамическим изменениям в процессе

управления разработкой системного ПО. Поэтому данные модели целесообразно использовать при моделировании отдельных компонент или структурных элементов интеллектуальных систем принятия решений или использовать в основе процесса выработки практических рекомендаций менеджерам.

Исследования моделей на основе аппарата управляемых случайных процессов позволили определить два характерных направления управления, реализуемых данным подходом: информационная невыводимость и информационное невлияние [5].

Среди достоинств данного подхода моделирования следует выделить возможность ее реализации в компьютерных системах в виде монитора ссылок и системы аудита.

Кроме того в сложных, многофакторных системах использование данного подхода моделирования позволяет анализировать отдельные составляющие без опасности снижения точности результатов моделирования в целом.

В то же время математические модели процесса разработки программного обеспечения в связи с учетом особенностей современной методологии SCRUM и факторов повышения требований к безопасности нуждаются в ряде доработок.

В автоматной модели управления технология разработки системного ПО представляется детерминированным автоматом, на вход которого поступает последовательность команд пользователей. Основными элементами автоматной модели являются: множество состояний системы  $\{V\}$ , множество пользователей  $\{U\}$ , множество матриц доступов  $\{M\}$ , множество команд пользователей, изменяющих матрицу доступов  $\{CC\}$ , множество команд пользователей, изменяющих состояние  $\{VC\}$ , множество выходных значений  $\{Out\}$ .

Среди достоинств данного подхода моделирования следует отметить многообразие подходов управления, определяющих не только правила распределения задач, но и конфигурацию, порядок взаимодействия между объектами и субъектами процесса разработки ПО.

Среди недостатков автоматных моделей можно отметить сложность их практической реализации в случае учета всего разнообразия подходов и методологий разработки системного ПО. Кроме того, задача учета фактора безопасности данным направлением моделирования также не решена.

Как отмечено в ряде источников [6], доминирующим при решении широкого круга задач анализа и синтеза систем управления различного назначения длительное время оставался графокомбинаторный подход. В этом случае процесс разработки ПО представляется в виде функции:  $G(N, C)$  или  $G(x)$ , где

$N$  – множество состояний в управлении,  $C$  – множество связей между состояниями,  $x$  – характеристика качества управления (эффективность, безопасность, стоимость и др.), выбранная в качестве критерия оптимизации. Тогда частная задача разработки системного ПО может трансформироваться в оптимизационную задачу вида:  $G(x) \rightarrow opt$ .

Решение сетевых задач управления в рамках данного подхода основывалось на моделировании процесса в виде графа, и было связано с направленным перебором возможных вариантов с целью достижения некоторого оптимума относительно анализируемого свойства.

На рис. 4 представлена сравнительная характеристика наиболее известных подходов математической формализации процессов управления разработкой с указанием их достоинств и недостатков.



Рис. 4. Сравнительная характеристика наиболее известных подходов математической формализации процесса управления разработкой системного ПО

Таким образом, в результате анализа и сравнительных исследований существующих моделей процесса разработки системного ПО были выявлены ряд характерных особенностей, достоинств и недостатков существующих направлений анализа и синтеза этих систем.

Исследование основных подходов моделирования показал, что большинство моделей, связанных с реализацией технологии разработки ПО не учитывают факторов априорной неопределенности в параметрах безопасности.

Кроме того, отсутствие учета в моделях динамических изменений в ходе разработки ПО (особенно-стей SCRUM) требует соответствующих исследований и разработок.

## Выводы

Проведен анализ основных требований к качеству программного обеспечения компьютерных систем критического применения, указано на приоритетность требований безопасности системного ПО и обязательность соблюдения этих требований на всех этапах жизненного цикла ПО.

Проведены исследования основных методов и средств обеспечения заданных требований. Рассмотрены существующие методологии разработки ПО, указано на недостаточность внимания со стороны разработчиков вопросам безопасности.

Проанализированы основные направления и подходы математического моделирования, выделены перспективные направления математической формализации процессов управления разработкой системного ПО.

Указано на целесообразность совершенствования существующих методологий разработки системного ПО, путем разработки новой технологии с учетом повышенных требований безопасности.

## Список литературы

1. Департамент QA: Ошибки управления [Электронный ресурс] – Режим доступа: <http://blog.alsedi.com/departament-qa-oshibki-upravleniya/>.
2. ДСТУ ISO/IEC 25012:2016 Інженерія систем і програмних засобів. Вимоги до якості систем і програмних

засобів та її оцінювання (SQuaRE). Модель якості даних (ISO/IEC 25012:2008, IDT) [Електронний ресурс] – Режим доступу:

[http://online.budstandart.com/ua/catalog/doc-page.html?id\\_doc=69135](http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69135).

3. Казарин О.В. Безопасность программного обеспечения компьютерных систем : Монография/ О.В. Казарин. – М.: МГУЛ, 2003. – 212 с.

4. Лунаев В.В. Надежность и функциональная безопасность комплексов программ реального времени / В.В. Лунаев. Институт системного программирования Российской академии наук – М: 2013. 176 с.

5. Семенов С.Г. Модель оценки риска разработки программного обеспечения. / С.Г. Семенов, Д.А. Лисица // Материалы XV Международной НТК «Проблемы информатики и моделирования». – X: НТУ «ХПИ», 2015. С.82.

6. Семенов С.Г. Методика математического моделирования защищенной ИТС на основе многослойной GERT-сети / С.Г. Семенов Методика математического моделирования защищен // Вестник Национального технического университета «Харьковский политехнический институт». Сборник научных работ. Тематический выпуск: Информатика и моделирование. – X: НТУ «ХПИ», 2012. – №62 (968). – С 173-181.

7. EXECUTIVE SUMMARY AGILITY ACCELERATES THE DELIVERY OF BUSINESS VALUE [Электронный ресурс] – Режим доступа:

<http://www.agile247.pl/wp-content/uploads/2017/04/versionone-11th-annual-state-of-agile-report.pdf>.

8. OSSTMM 3. Open-Source Security Testing Methodology Manual [Электронный ресурс] – Режим доступа: <http://www.isecom.org/mirror/OSSTMM.3.pdf>

9. Test & Test Case Management in Jira [Электронный ресурс] – Режим доступа:

<http://blog.alsedi.com/test-test-case-management-in-jira-part-0/>

10. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities, Proceedings of National Academy of Sciences, vol. 79 no. 8 pp. 2554–2558, April 1982.

Надійшла до редколегії 15.08.2017

Рецензент: д-р техн. наук, проф. С.Г. Семенов, Національний технічний університет «ХПИ», Харків.

## АНАЛІЗ І ДОСЛІДЖЕННЯ МОДЕЛЕЙ І МЕТОДІВ РОЗРОБКИ СИСТЕМНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кассем Халіфі, Т.М. Шипова

У статті проведено аналіз і дослідження моделей і методів розробки системного програмного забезпечення комп'ютерних систем критичного застосування. Представлені фактори, що впливають на це процес. Визначено вимоги до якості розробки системного програмного забезпечення в цілому і вказано на пріоритетність показників безпеки програмного забезпечення. Проведено аналіз і порівняльне дослідження основних підходів математичного моделювання процесу розробки системного програмного забезпечення. Визначено пріоритетні напрямки подальшого дослідження.

**Ключові слова:** системне програмне забезпечення, вимоги до якості програмного забезпечення, безпеку, SCRUM, Agile, математичне моделювання.

## ANALYSIS AND INVESTIGATION OF MODELS AND METHODS OF SOFTWARE SOFTWARE DEVELOPMENT

Kassem Khalife, T.M. Shipova

The article analyzes and studies models and methods for developing system software for critical computer systems. Factors affecting this process are presented. The requirements to the quality of the development of the system software in general have been determined and the priority of the software security indicators has been indicated. The analysis and comparative study of the main approaches of mathematical modeling of the system software development process are carried out. Priority directions for further research are determined.

**Keywords:** system software, software quality requirements, security, SCRUM, Agile, mathematical modeling.