

Л. А. Чижикова

**СИСТЕМНЫЙ АНАЛИЗ И СОЗДАНИЕ МЕТОДОЛОГИИ
РАЗРАБОТКИ ПРОГРАММНОГО ПРОДУКТА
МАТЕМАТИЧЕСКОЙ МОДЕЛИ ФИЗИЧЕСКИХ ПРОЦЕССОВ
ДЛЯ АВИАЦИОННОЙ ОТРАСЛИ**

37

Цель работы – обобщение существующего опыта и подходов к разработке программного обеспечения и написание методики создания программного продукта для применения в авиационной отрасли. В ходе работы были использованы теоретические методы исследования для выявления подходов к математическому описанию моделей атмосферных явлений сдвига ветра, турбулентности и эмпирические методы для определения оптимальной методологии разработки программных продуктов на основе собственного опыта автора, полученного в процессе работы на российских предприятиях авиационно-космической отрасли.

The purpose of this work was to summarize the existing experience and approaches to software development and create the methodology for a software product development for the aviation industry. The current research was based on the systemic analysis and the theory of statistical estimation. This work primarily deals with theoretical methods of research which were used to identify approaches to the mathematical description of atmospheric phenomena models of wind shear, turbulence. The author also analyses relevant the optimal methodology (Russian and foreign papers) for developing software products which takes into account the authors own experience obtained in the process of work at the Russian aerospace industry enterprises.

Ключевые слова: методология разработки ПО, сдвиг ветра, турбулентность, аэродинамика, математическое моделирование, авиационный тренажер, ПО для авиации.

Keywords: wind shear, turbulency, aerodynamics, mathematical modelling, software, flight simulation, aviation software.

Введение

Программная инженерия – довольно молодая наука, обладающая нишами для исследований и возможностями трансформации существующих подходов к созданию программных продуктов и унификаций [1–3].

Любая методика – технология, выработанная в результате успешного применения теоретической и статистической базы, она подразумевает комплекс необходимых действий для достижения конечного результата.

Что касается методологий создания программных продуктов, ситуация неоднозначна. Как отмечает М. Шоу, программная инженерия, в отличие от таких наук, как медицина, не имеет руководства по прове-



дению исследований для начала разработки. Исследования в этой области предназначены для улучшения практики разработки программного обеспечения (ПО) [4]. В целом считается, что «разработка алгоритма и составление программного кода для ЭВМ – творческий и трудно формализуемый процесс» [5, с. 137].

Рассматривая частный случай разработки программного обеспечения – имитации физических процессов, мы сталкиваемся с потребностью определения математической логики их работы, не считая традиционного применения общепринятых моделей управления жизненным циклом ПО и понимания процесса проектирования программных продуктов.

В публикации НАСА (NASA), посвященной проблематике применения наиболее популярного на сегодняшний день метода разработки ПО – гибкой методологии (agile methods), отмечено, что ее применение требует значительных корректировок для разработки критических по безопасности программных продуктов аэрокосмической отрасли [6].

В связи с большим числом авиакатастроф, вызванных атмосферными явлениями и их воздействием на аэродинамику полета летательного аппарата (ЛА), а также недостаточным количеством исследований данной проблемы и путей повышения безопасности полетов при воздействии сдвига ветра, турбулентности и явления микровзрыва, задача создания программного обеспечения математической модели данных явлений является актуальной [7].

Таким образом, статья посвящена актуальной на сегодняшний день проблеме методологии создания программных продуктов математических моделей для применения в авиационной отрасли.

Основное содержание исследования составляет анализ существующих подходов к созданию программного обеспечения и математического моделирования. Значительное внимание уделяется специфике разработки программных продуктов для авиационной отрасли.

Цель исследования – выявить и сформулировать подход к разработке ПО математических моделей таких атмосферных явлений, как сдвиг ветра, турбулентность и микровзрыв, применительно к авиационной отрасли.

В ходе работы рассматриваются наиболее известные подходы к разработке программного обеспечения, анализируется опыт математического моделирования атмосферных явлений, изучаются статистические данные разрабатываемых программных продуктов и проблем данной области. На их основании формулируется методика разработки программного обеспечения математических моделей физических процессов для применения в авиационной отрасли.

Основная часть

Методология, как и методика в широком понимании, представляет технологию выполнения операций для достижения конкретного положительного результата, она направлена на рекомендацию наиболее



эффективных, рациональных вариантов и образцов действий применительно к определенному виду деятельности. Как правило, данная технология вырабатывается в результате успешного применения операций и достижения положительного результата.

На протяжении всего периода становления программной инженерии как науки прослеживается потребность в поиске новых методов создания программных продуктов. Впервые термин «кризис программного обеспечения» был введен на конференции НАТО «Инженерия программного обеспечения» в 1968 г. [8]. Он подразумевал проблему быстрого роста вычислительной мощности и сложность задач, которые невозможно решить. С повышением сложности программного обеспечения из-за отсутствия четких методов возникли многие проблемы в данной сфере [8].

39

Статистика разрабатываемых программных продуктов

Согласно статистике, приведенной в публикации американского авиационного издания *Aviation Today*, при создании бомбардировщика-истребителя *Lokheed Martin F-35* требуется перерасход первоначально запланированного бюджета из-за некорректно спроектированного программного обеспечения [9].

В апреле 2017 г. Государственное подотчетное ведомство США опубликовало 49-страничный отчет, в котором отмечалось, что задержки с тестированием бомбардировщика-истребителя *F-35* могут стоить оборонному департаменту дополнительный миллиард долларов США, не считая затрат на приобретение, которые уже составили 400 млрд [10]. Проблемы заключались в основном программном модуле – *Block 3F*, который обеспечивает обмен информацией по каналам связи, взаимодействие с другими программными блоками, управление вооружением и предполетную подготовку.

Помимо разработки программных продуктов для авиационной отрасли, требующих соответствия определенным критериям безопасности, предпосылками создания российских программных продуктов стали возрастающая потребность в автоматизации производственных процессов, импортозамещении, создании инструментов для достижения исследовательских, научных и производственных целей.

По мнению авторов обзора 20 российских операционных систем, размещенного на сайте Ассоциации разработчиков программных продуктов «Отечественный софт» [11; 12], «существует два подхода к созданию российского софта. Первый заключается в написании исходного кода продуктов с нуля, полностью силами отечественных специалистов. Второй вариант предполагает создание национального ПО на основе доработки заимствованных исходных кодов. Именно его и придерживаются работающие на ниве импортозамещения ПО российские софтверные компании» [12].

При создании программных продуктов для авиационной отрасли корректировка заимствованного программного кода ведет к его регрес-

сии, что впоследствии требует дополнительных затрат на доработку программного продукта и является некорректным и неадекватным способом разработки программного обеспечения.

По данным Министерства цифрового развития, связи и массовых коммуникаций РФ, на конец марта 2018 г. в Едином реестре российских программ для электронных вычислительных машин и баз данных зарегистрировано 4346 отечественных программных продуктов [13]. На 17 октября 2016 г. среди программных продуктов, создаваемых в РФ, наибольшую долю занимает прикладное программное обеспечение (78 %) для решения специфических задач отдельных отраслей производства. «К этому классу правообладатели отнесли 1259 своих продуктов. Среди системного ПО лидируют средства обеспечения информационной безопасности (263 продукта)» [14].



Рис. 1. Содержание реестра российского ПО [14]

Программное обеспечение, являющееся инструментом исследований и автоматизации, можно отнести к двум категориям — системному и прикладному. Потребность в решении специфических отраслевых задач с помощью ПО подтверждает статистика. Однако программный продукт математической модели атмосферных явлений нельзя отнести к прикладному программному обеспечению, поскольку его характеристики не соответствуют этой категории: данный вид ПО предполагает использование в составе изделия — программно-алгоритмического комплекса авиационного тренажера.

Таким образом, мы видим не вполне однозначную интерпретацию процесса разработки ПО, а также потребность в создании и апробации новых методик.

Анализ подходов к созданию программного обеспечения

С позиции оперативного управления проектами могут быть выбраны различные бизнес-процессы для достижения эффективного результата разработки — программного продукта. Сейчас наиболее известными и часто применяемыми подходами к разработке ПО являются



гибкая методология разработки (agile methods), набор принципов для разработки сложного ПО в условиях невозможности конечного определения требований (SCRUM), методика экстремального программирования (XP), технология бережливого производства (lean manufacturing).

Так, философия бережливого производства, направленная на снижение затрат на всех этапах разработки, также применима к процессам разработки программных продуктов.

Создание программного модуля математической модели физических процессов можно рассматривать как ситуацию неопределенности, в которой предпочтительнее всего использовать подход SCRUM. Согласно исследованиям компании «Руссофт», доли компаний, использующих SCRUM или его аналог, в 2015 и 2016 гг. оказались одинаковыми – 13 % [15].

Как отмечено в публикации НАСА, для разработки критического по безопасности ПО, используемой в авиакосмической отрасли, применяют гибкую методологию. Команда разработки программного обеспечения Космического центра Маршалла (MSFC) использует данную методологию в течение последних двух лет, совмещая адаптированные гибкие методы с более традиционными [6].

Несмотря на проблемы в применении гибкой методологии, ее развитие привносит творческий подход и открывает новые возможности миссиям следующего поколения НАСА. Кроме того, многие коммерческие партнеры НАСА используют или будут использовать именно гибкую методологию разработки программных продуктов в своих проектах [6].

При этом применение таких традиционных подходов, как каскадная модель (waterfall model), основанная на последовательном планировании операций разработки ПО, в современных компаниях не упоминается [6].

Самое большое различие между гибкими методами и традиционными, основанными на планировании, заключается в том, что гибкие методы не ограничивают разработку сроками, неизменностью составленных требований [6].

Технологический процесс – полная модель жизненного цикла – гибкой методологии предусматривает следующие стадии [16]:

- определение концепции проектов разработки и приоритетов их выполнения;
- идентификация членов команды, финансирования, начальных условий и требований;
- итерационный подход к разработке, который повторяется на протяжении всего жизненного цикла;
- тестирование программного продукта, разработка документации поддержка использования разработанного программного обеспечения;
- закрытие поддержки разработанного программного продукта.

В жизненном цикле гибкой методологии доминирует фаза итеративного процесса. Каждая итерация приводит к следующей части «головломки» разработки программного обеспечения, пока окончательный программный продукт не будет получен.



Итеративный процесс можно разделить на следующие этапы:

- определение требований к программному продукту;
- разработка программного обеспечения;
- тестирование, внутреннее и внешнее обучение использованию программного продукта;
- интеграция;
- обратная связь (выполнение следующей итерации по разработке программного продукта на основе отзывов о его использовании).

Вышеупомянутые подходы рассматривают программное обеспечение как отдельный продукт, причем преимущественно с точки зрения бизнес-требований. В этом случае при планировании работ часто не закладываются такие аспекты детализации разработки, как формализация функциональных требований программного продукта, аналитика логики и математических описаний, происходящих в программе.

Также данные подходы не учитывают специфику какой-либо отрасли и ее специфических требований – безопасности или отказоустойчивости. На глубокое понимание и детализацию разработки, направленную на исключение возможных ошибок в работоспособности и функционировании программы, ни один из вышеуказанных методов не рассчитан, вследствие чего возникает потребность в корректировке и разработке отдельной методологии.

Принципы построения математической модели

Математические модели – обширный класс знаковых моделей, отражающих существенные черты объекта реального мира посредством математического описания.

Согласно рекомендациям стандарта NASA-STD-7009 [17], выработанного с учетом специфики разработки программного обеспечения для авиационной отрасли, при создании программного продукта математической модели выделяется 11 этапов разработки (табл. 1).

Таблица 1

Этапы разработки математической модели программного продукта

Этап процесса моделирования и симуляции	Описание этапа	Важность этапа
Представление системы в реальном мире (Real World System)	Определение объекта / субъекта моделирования/ анализа	Система представления в реальном мире (RWS) находится в центре внимания моделирования и симуляции. Из аспектов системы модель разрабатывается, по этому направлению также происходит анализ на протяжении всей разработки



Продолжение табл. 1

Этап процесса моделирования и симуляции	Описание этапа	Важность этапа
Исследование системы (Examine the System)	Декомпозиционный анализ предмета моделирования / симуляции с целью определения соответствующих деталей – аспектов для включения в модель	Определяет абстракции (что включать в модель), уровень детализации, положения и составные взаимосвязи модели
Концептуальная модель и спецификация модели (Conceptual Model and Model Specification)	Сбор абстракций, положений и описаний физических процессов, представляющих поведение моделируемого объекта в реальности, из которых могут быть созданы математическая модель или валидационные эксперименты	Представляет текстовые описания, блок-схемы, тематические описания, уравнения или псевдокоды, которые, в свою очередь, обеспечивают статистическое представление системы реального мира, дополняемое спецификацией модели
Валидация концептуальной модели (Validation of Conceptual Model)	Процесс определения степени, в которой концептуальная модель является точным представлением реального мира с точки зрения предполагаемого использования модели или симуляции	Удостоверяет, что существенные аспекты представления системы реального мира учтены для предстоящего анализа и создания модели / симуляции
Имплементация модели (Model Implementation)	Создание моделей (для компьютерных моделей это означает кодирование алгоритмов), которые будут представлять собой поведение системы	Модель системы реального мира создается в соответствии со спецификацией
Верификация вычислительной / симуляционной модели с концептуальной (Verification of Computational / Simulation Model with Conceptual Model)	Процесс определения того, что концептуальная модель является точным представлением системы в реальном мире из перспектив предполагаемого использования продукта модели или симуляции	Проверяется, что модель создана в соответствии со спецификацией
Вычислительная модель / симуляция (Computational / Simulation Model)	Компьютерный код и логика, включая числовые представления математической модели, которые имитируют характеристики системы, сущность явления или процесс	Вычислительная / имитационная модель является основой для анализа соответствия системе реального мира



Окончание табл. 1

Этап процесса моделирования и симуляции	Описание этапа	Важность этапа
Валидация вычислительной модели / симуляции (Validation of Computational / Simulation Model with RWS or Analogous Referent)	Процесс определения, в какой степени вычислительная / имитационная модель является точным представлением реального мира в зависимости от перспективы предполагаемого использования модели / симуляции	Проверяет, что модель адекватно представляет предполагаемую систему реального мира
Разработка и запуск сценариев работы системы (Develop and Run Scenarios (Design of Experiments))	Описание или определение соответствующей системы, внешних воздействий (воздействия внешних сред), условий и / или параметров (входные данные для модели / симуляции), используемых для управления ходом событий во время моделирования / симуляции	Набор сценариев для запуска определяет полноту анализа и может поддерживать анализ чувствительности параметров сценария
Выходные данные моделирования и симуляции (M&S Output Data)	Данные, полученные посредством моделирования и симуляции при выполнении сценариев	Необработанные выходные данные, полученные от запущенных моделей и симуляций с интересующими сценариями-алгоритмами работы; могут быть далее обработаны для получения результатов, непосредственно связанных с анализом
Анализ результатов на предмет соответствия представлению реального мира (Results Analysis Applied to RWS)	Анализ выходных данных моделирования и симуляции, применимых к системе реального мира	Модель / симуляция проходит полный цикл на предмет соответствия системе реального мира

С точки зрения общепринятых методов разработки программных продуктов сложно отнести данную методологию к одному этапу жизненного цикла. Стандарт NASA-STD-7009 занимает особое место в мире моделирования и симуляции, поскольку эта методика, как правило, применима ко всем типам моделирования и симуляции и на всех этапах разработки.



В классическом понимании при создании математической модели выделяют следующие этапы [4]:

- изучение объекта / процесса, определение целей моделирования (для чего создается модель – понимание процессов, происходящих в ней, оптимизация объекта, подбора параметров, управления объекта (возможно, симуляция воздействия или устройства), прогнозирование поведения объекта или процесса);

- огрубение объекта / процесса (поскольку успех моделирования заключается в корректном определении характеристик модели, процесс огрубения предполагает ранжирование и удаление менее значимых факторов, способствует пониманию главных свойств и закономерностей модели);

- поиск математического описания (переход от абстрактного описания процесса / объекта к математическим формулировкам);

- математическая модель (формализация математического описания в виде системы уравнений, неравенств и т. п.);

- выбор метода исследования (решение задачи описываемой модели);

- разработка алгоритма, логики работы и программного кода (на основании разработанных описаний создается программный продукт математической модели);

- верификация, тестирование (запуск и отладка созданной программы);

- анализ результатов (анализ результатов на соответствие модели представлению в реальном мире);

- уточнение модели (при отрицательном результате моделирования и неполном соответствии модели представлению объекта / процесса в реальном мире производится уточнение и корректировка модели).

Данные этапы математического моделирования больше напоминают каскадный, последовательный подход к планированию разработки (определение требований к системе и программному обеспечению, анализ, определение архитектуры ПО, кодирование, тестирование, использование ПО). Но в силу аргументов, приведенных в пользу применения гибкой модели разработки ПО, данная работа фокусируется на разработке методики, основанной именно на таком подходе.

Исходя из вышеописанного, можно выделить три основных базиса формирования методики управления разработкой программного обеспечения математической модели физических процессов: применение гибкой методологии, разработка согласно этапам, предложенным стандартом NASA-STD-7009, учет основных аспектов создания математической модели.

Определение процессов атмосферных явлений ветра

В авиационной отрасли одной из немаловажных задач наряду с разработкой системного программного обеспечения для бортовых машин является обучение пилотированию на авиационных тренажерах.



В общем случае реализация в программном модуле такой математической модели, как внешние атмосферные воздействия, дает возможность создания инструмента для исследования физических процессов воздействия таких атмосферных явлений на объекты климатологии.

Согласно рекомендациям по повышению уровня безопасности полетов, необходима подготовка (тренировка) пилотов на знание навыков управления ЛА в условиях сдвига ветра [18]. Рекомендации по разработке бортового программного обеспечения для обнаружения сдвига ветра изложены в стандартах RTCA DO-178 [19] и AC25-12 [20]. Однако определенных рекомендаций для разработки программного обеспечения математической модели сдвига ветра и турбулентности для авиационных тренажеров нет, ввиду чего исследования такого рода необходимы.

Из всех атмосферных явлений и метеорологических факторов, влияющих на полет воздушного судна (ВС), ветер является одним из наиболее важных, поскольку вызывает дополнительное перемещение ВС относительно Земли.

В зарубежных работах для симуляции таких явлений предлагались различные подходы [21–23]. В российском опыте создания математических моделей сдвига ветра, турбулентности и их воздействия на аэродинамику нет окончательного алгоритмического и математического описания для реализации в программном обеспечении авиационного тренажера.

Частным решением построения математической модели атмосферных явлений ветра является параметрическое воспроизведение в программном модуле статистических данных, описанных в ГОСТ 24728-81 Ветер. Пространственное и временное распределение характеристик [24]. Но данное представление не полностью учитывает воздействие воздушных масс на аэродинамику ЛА, а также не предоставляет дальнейшей возможности моделирования маневрирования и предотвращения воздействия таких атмосферных явлений на ВС.

Воздействия воздушных масс, часто являющиеся причинами авиакатастроф, можно разделить на сдвиги ветра и турбулентность. Также в ряде работ рассматривается так называемый микровзрыв – атмосферное явление, возникающее в результате столкновений холодного и теплого атмосферных фронтов и характеризующееся нисходящим движением воздушных масс [21; 25].

Как правило в атмосфере постоянно присутствует движение воздушных масс – горизонтальный ветер, характеризующийся скоростью и направлением.

Согласно «Руководству по прогнозированию метеорологических условий для авиации», сдвиг ветра – атмосферное явление, при котором резко изменяется его скорость и направление в пространстве, включая восходящие и нисходящие воздушные потоки [26].

При применении инженерного математического пакета программ Matlab для вычисления скорости сдвига ветра используется блок, графически представленный на рисунке 2.

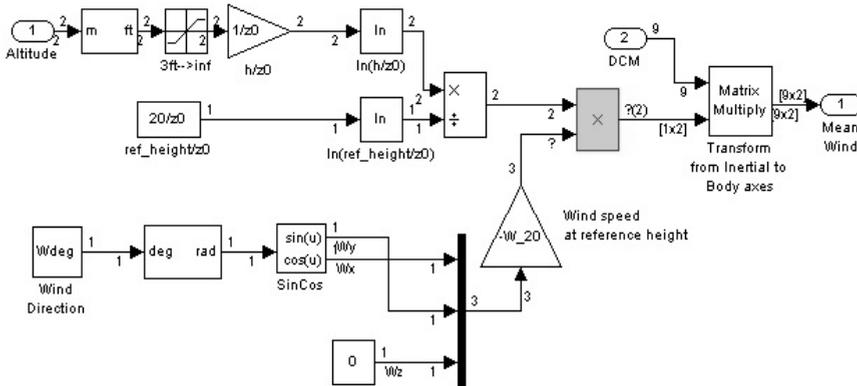


Рис. 2. Графическое представление математической модели сдвига ветра

Чаще всего в математическом описании скорости сдвига ветра в зависимости от высоты полета (эшелона) используется уравнение Прандтля, известное также как логарифмический закон ветра:

$$u_w = W_{20} \frac{\ln\left(\frac{h}{z_0}\right)}{\ln\left(\frac{20}{z_0}\right)},$$

где u_w – средняя скорость ветра; W_{20} – измеренная скорость ветра на высоте в 20 футов; h – высота; z_0 – константа, зависящая от фазы полета.

Данное выражение используется при вычислениях скорости сдвига ветра в математическом пакете Matlab в соответствии с рекомендациями зарубежного стандарта MIL-F-8785 [27].

Обычно атмосферное явление турбулентности принято описывать как вихревые движения воздушных масс. Н. А. Воронцов дает следующее определение: «Турбулентность, вызывающую болтанку самолетов, можно представить как пульсации в пространстве восходящих и нисходящих потоков различных поперечных сечений (масштабов), соизмеримых с размерами самолета, без заметного изменения высоты полета» [28, с. 20].

При попадании самолета в зону турбулентности возникает кратковременный вывод ВС из равновесия.

Для симуляции и математического описания турбулентности и пульсаций скоростей ветра используются различные методики. Наиболее распространенной является симуляция посредством подачи на вход математической модели возмущения белого шума по Гауссу, но данный метод не соответствует действительности, в том числе процессу турбулентных движений, происходящих в атмосфере [29].



Математическое описание турбулентности берет свое начало от уравнений Навье – Стокса, закона Рейнольдса и уравнений Блазиуса. Уравнения Навье – Стокса ввиду их нелинейности на практике неприменимы. Поэтому для симуляции процесса турбулентных течений воздушных масс, как правило, используют уравнения Теодора фон Кармана [30].

Микропорывы, обусловленные нисходящими потоками воздушных масс, или микровзрывы ветра – широко известное атмосферное явление, предмет исследования новаторской работы Т. Фуджиты. Фуджита определяет нисходящий поток как краткосрочное сильное сокращение, которое вызывает сильно расходящуюся вспышку. Микропорывы же определяются как мелкомасштабный нисходящий поток воздуха с его выбросами и ветрами, простирающимися примерно на 4 километра или менее [31].

При моделировании микропорывов логика работы данного процесса может быть представлена в виде трехмерного вихревого поля, в котором выделяется тороидальная область (ядро), где скорость, начиная от нулевой в центре, линейно увеличивается по радиусу до границы ядра. Вне ядра вихревое поле задается функцией потока через приближенно вычисляемые комбинации полных эллиптических интегралов. Численное дифференцирование функции потока дает радиальную и вертикальную составляющие скорости ветра. Первая из них раскладывается на две компоненты: параллельно и перпендикулярно взлетно-посадочной полосе [25].

Определение основных аспектов методологии разработки

Для определения оптимальной методологии разработки программного обеспечения математической модели физических процессов для аэрокосмической отрасли в качестве базиса рассматривается модель жизненного цикла программного обеспечения, в данном случае – гибкая. Далее для каждого этапа определяется специфика – техническая и математическая детализация с точки зрения подходов к построению математического моделирования и симуляции с учетом требований к разрабатываемому программному продукту на основе нормативных документов отрасли.

В программной инженерии под разработкой, помимо кодирования алгоритмов, понимается комплекс действий, направленных на получение программного продукта.

Другой немаловажный этап разработки ПО – обеспечение условия модульности его архитектуры. Поэтому описание логики и программного кода каждого явления вынесем в отдельный модуль [32; 33].

Сравнительная матрица этапов разработки программного обеспечения представлена в таблице 2.

Таблица 2

Сравнительная матрица этапов разработки программного обеспечения

Этапы разработки (согласно принципам гибкой методологии)	Техническая детализация	Этапы разработки (в соответствии со стандартами надежности HDBK)
Определение концепции проектов разработки и приоритетов их выполнения (Concept)	–	–
Идентификация членов команды, финансирования, начальных условий и требований (Insertion)	Исследование исходного объекта / процесса	Анализ реальной системы, ее исследование, определение процессов
	Определение целей моделирования	
	Огрубление	
Requirements	Поиск математического описания	Декомпозиционный анализ (исследование) предмета моделирования / симуляции
	Математическая модель (спецификация)	
	Выбор метода исследования	
Итерационный подход к разработке, который повторяется на протяжении всего жизненного цикла (Iteration / Construction)	Development	Верификация вычислительной / симуляционной модели с концептуальной
	Testing	Валидация концептуальной модели
Тестирование программного продукта, разработка документации (Release)	Delivery	Имплементация модели
	Feedback	Верификация вычислительной / симуляционной модели с концептуальной
		Вычислительная модель / симуляция
Поддержка использования разработанного программного обеспечения (Production)		Валидация вычислительной модели / симуляции
		Разработка и запуск сценариев работы системы
		Выходные данные моделирования и симуляции
Закрепление поддержки разработанного программного продукта (Retirement)	Отладка и тестирование программы, расчеты, анализ результатов, уточнение модели	Анализ результатов на предмет соответствия сравнению с реальным миром



Основные этапы предлагаемой методологии:

- 1) определение концепции проектов разработки и приоритетов их выполнения;
- 2) исследование исходного объекта / процесса;
- 3) определение целей моделирования;
- 4) анализ реальной системы, ее исследование, определение процессов;
- 5) идентификация членов команды, финансирования;
- 6) выполнение итераций на основе подхода гибкой методологии до достижения положительного результата разработки;
- 7) тестирование программного продукта, составление документации;
- 8) прекращение доработок и сервиса программного продукта.

Итерационный цикл включает в себя выполнение следующих этапов:

- 1) декомпозиционный анализ, исследование предмета моделирования/симуляции;
- 2) огрубение;
- 3) поиск математического описания;
- 4) создание концептуальной математической модели и ее спецификации;
- 5) определение архитектуры ПО;
- 6) разработка детальных алгоритмов работы моделируемой системы, программного кода;
- 7) верификация вычислительной/симуляционной модели с концептуальной;
- 8) разработка и запуск сценариев работы системы;
- 9) отладка и тестирование программы, расчеты, анализ результатов, уточнение модели;
- 10) валидация вычислительной модели/симуляции;
- 11) анализ результатов на предмет соответствия представлению реального мира.

Отметим, что сложность применения гибкой методологии для разработки программного обеспечения заключается в неизменности нормативных требований и рекомендаций создания архитектуры ПО.

Часто при изначально некорректно спроектированном архитектурном решении возникает проблема регрессии программного кода, когда дальнейшее развитие и сопровождение невозможны [32].

Поэтому в данной методике предполагается, что архитектура проектируется единойжды при понимании логики и процессов моделируемой системы, а также логики сопряжения с другими программными модулями сложной системы.

Заключение

Основная проблема создания корректно функционирующего программного продукта математической модели лежит в разрозненности направлений коммерческого и научного подходов к разработке продуктов.

Наряду с проектированием и пониманием архитектуры программного обеспечения ключом к созданию успешного программного продукта является как качественное управление разработкой, так и понимание действий для реализации системы, методики ее создания.



На ранних этапах проектирования для последующего определения алгоритма работы математической модели и программной реализации необходимо определить основные физические величины, их взаимодействие, а также параметрические составляющие и переменные программного продукта.

В зависимости от детализации моделируемой системы (определенной на этапе огрубения) можно создать библиотеки подключаемых статистических данных, а также подпрограммы – генераторы событий моделируемой системы.

Предложенная методика вкупе с математико-логическим описанием процессов атмосферных явлений сдвига ветра, нисходящих потоков и турбулентности позволяет реализовать программный продукт физических процессов в составе программного обеспечения авиационного тренажера для отработки навыков пилотирования. Также программная реализация данного алгоритма может служить для исследований влияния ветровых воздействий и турбулентности на аэродинамику ВС.

Описанный в статье алгоритм также направлен на создание общих отвлекающих факторов при обучении на авиационном тренажере для усложнения пилотирования летчиком, а также для исследования влияния на аэродинамику ВС.

Существующая классификация программных средств (ГОСТ Р ИСО/МЭК ТО 12182-2002) [34] не учитывает специфику каждой отдельной отрасли и ее требований к программным продуктам. Чаше всего вновь создаваемое программное обеспечение может использоваться как инструмент для достижения конечного результата на отдельных этапах жизненного цикла изделия – системы в отдельных отраслях производства. Поэтому при разработке необходимо учитывать специфические нужды и особенности этих отраслей. В настоящее время в авиационной отрасли существуют жесткие требования к сертификации и надежности создаваемого программного обеспечения. Дальнейшие исследования и аналитическая работа будут направлены на создание классификации типов программного обеспечения в авиационной отрасли и специфических требований к нему.

Список литературы

1. *Campbell-Kelly M.* Development and Structure of the International Software Industry // Business and Economic History. 1995. Vol. 24, №2. P. 73 – 110.
2. *Wirth N.* A Brief History of Software Engineering // IEEE Annals of the History of Computing. 2008. Vol. 30, iss. 3. P. 32 – 39.
3. *Position Papers for Dagstuhl Seminar 9635 on History of Software Engineering / eds. A. Brennecke, R. Keil-Slawik.* URL: <https://www.dagstuhl.de/Reports/96/9635.pdf> (дата обращения: 17.05.2018).
4. *Shaw M.* What Makes Good Research in Software Engineering? // International Journal of Software Tools for Technology Transfer. 2002. Vol. 4, iss. 1. P. 1 – 7.
5. *Ясницкий Л.Н., Данилевич Т.В.* Современные проблемы науки : учеб. пособие. М., 2017.
6. *Agile Development Brings New Challenges for Software Assurance at NASA // NASA Office of Safety & Mission Assurance Website.* URL: <https://sma.nasa.gov/news/articles/newsitem/2014/09/04/agile-development-brings-new-challenges-for-software-assurance-at-nasa> (дата обращения: 07.03.2018).



7. Обидин Н.И., Буран Т.Р. Влияние сдвига ветра на безопасность полетов и пути ее повышения // Системы обработки информации. 2015. №3. С. 144–146.
8. Naur P., Randell B. Software Engineering: Report on a Conference Sponsored by the NATO Science Committee. Brussels, 1969.
9. Bellamy W. III New Approaches to Developing Avionics Software // Avionics International. URL: <http://interactive.aviationtoday.com/new-approaches-to-developing-avionics-software/> (дата обращения: 16.04.2018).
10. GAO-17-351. F-35 Joint Strike Fighter. DOD Needs to Complete Developmental Testing Before Making Significant New Investments. 2017.
11. Made in Russia: обзор 20 российских операционных систем. Часть 1 // АРПП «Отечественный софт»: [сайт]. URL: <http://www.arppsoft.ru/news/review/8760/> (дата обращения: 10.02.2018).
12. Made in Russia: обзор 20 российских операционных систем. Часть 2 // АРПП «Отечественный софт»: [сайт]. URL: <http://www.arppsoft.ru/news/review/8764/> (дата обращения: 10.02.2018).
13. В Минкомсвязи состоялось заседание Экспертного совета по российскому ПО // Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации: [офф. сайт]. URL: <http://minsvyaz.ru/ru/events/38106/> (дата обращения: 11.02.2018).
14. Статистические данные Министерства связи Российской Федерации // Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации: [офф. сайт]. URL: <http://minsvyaz.ru/ru/events/35881/> (дата обращения: 19.02.2018).
15. Экспорт российской индустрии разработки программного обеспечения: результаты исследования софтверной области по регионам // НП «РУССОФТ»: [сайт]. URL: <http://www.russoft.ru/report/4424> (дата обращения: 25.02.2018).
16. Ambler S., Lines M. Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise. URL: <http://ptgmedia.pearsoncmg.com/images/9780132810135/samplepages/0132810131.pdf> (дата обращения: 22.05.2018).
17. NASA-STD-7009. Standard for Models and Simulations. 2013.
18. Об утверждении федеральных авиационных правил «Подготовка и выполнение полетов в гражданской авиации Российской Федерации»: приказ Министерства транспорта Российской Федерации от 31.07.2012 г. №128. Доступ из справ.-правовой системы «Гарант».
19. RTCA DO-178C. Software Considerations in Airborne Systems and Equipment Certification. 2011.
20. AC №20-115C. Airborne Software Assurance. 2013.
21. NASA Technical Paper 2886 DOT/FAA/DS-89/06. Piloted Simulation. Evaluation of Recovery. Guidance for Microburst Wind Shear Encounters. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19890007449.pdf> (дата обращения: 11.04.2018).
22. NASA Conference Publication 2474. Wind Shear / Turbulence. Inputs to Flight. Simulation and Systems / eds. R. Bowles, W. Frost. URL: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19870015834.pdf> (дата обращения: 14.04.2018).
23. Barr N. Wind Models for Flight Simulator Certification of Landing and Approach Guidance and Control Systems Boeing. URL: https://archive.org/details/DTIC_ADA003801 (дата обращения: 23.03.2018).
24. ГОСТ 24728-81. Ветер. Пространственное и временное распределение характеристик. М., 1982.
25. Боткин Н.Д., Пацко В.С., Турова В.Л. Разработка численных методов построения экстремальных ветровых возмущений, действующих на самолет на этапе посадки. Разработка алгоритмов построения экстремальных ветровых возмущений: отчет №01880003467/02880054701 (ВНТИЦ). Свердловск, 1987. URL: <http://sector3.imm.uran.ru/rapevv1987/index.html> (дата обращения: 28.02.2018).



26. *Руководство по прогнозированию метеорологических условий для авиации* / под ред. К.Г. Абрамович, А.А. Васильева. Л., 1985.
27. MIL-F-8785C Military Specification. Flying Qualities of Piloted Airplanes. 1980.
28. *Воронцов П.А. Турбулентность и вертикальные токи в пограничном слое атмосферы.* Л., 1966.
29. *Фабрикант Н.Я. Аэродинамика. Общий курс.* М., 1964.
30. *Moorthouse D., Woodcock R. Background Information and User Guide for MIL-F-8785C, Military Specification – Flying Qualities of Piloted Airplanes.* URL: <http://contrails.iit.edu/files/original/AFWALTR81-3109.pdf> (дата обращения: 14.03.2018).
31. *Ferrero E., Mortarini L., Manfrin M. et al. Physical simulation of atmospheric microbursts // J. Geophys. Res.: Atmos.* 2014. Vol. 119, iss. 11. P. 6292–6306.
32. *Чижикова Л.А. Принципы проектирования модульной архитектуры программного обеспечения авиационной тематики // Программные продукты и системы.* 2017. №2. С. 291–300.
33. *Martin M., Ishii K. Design for Variety: Developing Standardized and Modularized Product Platform Architectures // Res. Eng. Des.* 2002. Vol. 13, iss. 4. P. 213–235. doi: 10.1007/s00163-002-0020-2.
34. *ГОСТ Р ИСО/МЭК ТО 12182-2002. Информационная технология. Классификация программных средств.* М., 2002.

Об авторе

Людмила Андреевна Чижикова – генеральный директор ООО «Лаборатория научно-прикладных исследований и разработок для авиации», Россия.
E-mail: ludmilachizhikova@yahoo.com

The author

Ludmila A. Chizhikova, General Director of LLC Laboratory of Scientific and Applied Research and Development for Aviation, Russia.
E-mail: ludmilachizhikova@yahoo.com