

УДК 004.413.2

А.С. Баженов, В.М. Ицыксон

Применение принципов прогнозирования в задачах проектирования программного обеспечения

Прогнозирование процессов создания программного обеспечения является одним из важнейших вопросов в области промышленной разработки. Этот аспект находит отражение в различных методологиях и моделях зрелости, таких как СММИ, но не имеет универсальных способов решения. В рамках данного исследования рассматриваются современные тенденции многофакторного анализа процессов разработки ПО, а также предлагается подход формирования научных гипотез в области управления проектами по разработке, который должен увеличить предсказуемость протекающих в них процессов.

Ключевые слова: разработка программного обеспечения, модель, фактор, прогнозирование, СММИ.

Быстрое развитие области промышленной разработки программного обеспечения (ПО) спровоцировало появление огромного количества новых научно-технических проблем и задач. Возможность и востребованность их решений в индустрии в каждый момент времени определялись и определяются степенью развития дисциплины производства ПО, существующими аналитическими и математическими аппаратами и, безусловно, потребностями заказчиков и конечных потребителей программных продуктов. Если в 1970-х годах человечество пыталось сформировать базовые методики, позволяющие структурировать и взять под контроль процесс создания ПО, то сегодня данная проблема переросла скорее в задачу выбора, а также привела к необходимости упрощения формального процесса работы для увеличения его гибкости. Одной из важнейших проблем в современной разработке являются вопросы предсказуемости и прогнозируемости процесса проектирования и создания ПО, что находит отражение в существующих моделях, описывающих уровень зрелости деятельности различных организаций, например СММИ [1]. Многомиллионные программные проекты сегодня становятся типовыми, и как следствие заказчики ожидают гарантий по разумному использованию бюджета и сдачи проекта в срок, предъявляют дополнительные требования к качеству результатов.

Решение задач прогнозируемости и предсказуемости проектов разработки ПО любого уровня становится возможным с использованием современных средств и методик в областях управления проектами и построения моделей бизнес-процессов. Моделирование и последующий анализ с помощью существующего математического аппарата и вычислительных средств позволит упростить принятие управленческих и технологических решений, оценить характеристики проекта разработки и его результатов, предсказать появление тех или иных проблем в будущем и многое другое. Таким образом, совмещение таких понятий, как проектирование, бизнес-процесс и моделирование, становится не только решаемой, но и исключительно актуальной задачей в области промышленной разработки ПО.

Работа организована следующим образом. В первом разделе рассказывается об области исследования и существующих тенденциях в рамках управления проектами разработки ПО. Второй раздел содержит описание задачи прогнозирования, примеры ее решения и актуальность прогнозирования в области разработки ПО. Третий раздел посвящен описанию метамодели программного проекта и базового метода моделирования. В четвертом разделе формулируются дальнейшие шаги исследования.

Область исследования

Возможность спрогнозировать факты, характеристики или события в будущем всегда интересовала и продолжает интересовать человечество. Причем в современном мире вопрос о том, каким предстанет будущее, становится одним из наиболее важных. Он также актуален и в рамках современной науки и технологии [2].

1. Прогнозирование и разработка программного обеспечения

Собственно прогнозирование – это научное исследование перспектив в некоторой области, предметом изучения которого является будущее, а продуктом, результатом исследований выступают научные выводы о вероятном состоянии изучаемого объекта. Широкое распространение прогнозирование получило в экономической, социальной, политической, культурной сферах, а также в области управления проектами разработки ПО [2].

Разработка программного обеспечения – это род деятельности, направленный на создание и поддержание работоспособности программного обеспечения, используя технологии, методологии и практики из информатики, математики, инженерии, управления проектами и других областей знаний [3]. Термин «разработка ПО» обычно используют для обозначения процесса написания и поддержки исходного кода, в то время как его значение намного шире и включает в себя исследование, разработку, модификацию, реинжиниринг, поддержку либо другую активность, целью которой является программный продукт. Работа над программными продуктами в общем случае ведется командой в рамках проекта разработки ПО, что влечет за собой появление ряда других активностей по управлению, координации, формированию бюджета, оценки и выработки критериев качества, анализу рисков и проблем и многому другому. Целью большинства проектов является достижение заданного с помощью ряда параметров (качественных и количественных) результата в отведенный срок и бюджет.

Совмещение понятий прогнозирования и разработки ПО открывает новые возможности для более эффективного управления процессом создания программных продуктов. Такой результат достигается в частности благодаря тому, что прогнозирование позволяет решить либо приблизиться к решению ряда существующих производственных проблем.

2. Проблемы разработки и прогнозирования

Использование прогнозирования в области разработки программного обеспечения позволяет делать процесс создания продукта более детерминированным, результаты более ожидаемыми, а принимаемые решения – эффективными. В частности, прогнозирование полностью решает или уменьшает негативное влияние ряда существующих в предметной области проблем [4, 5]:

- Недостаточная прозрачность процесса работы в проекте, не позволяющая оценить его текущее состояние, определить характеристики, например степень завершенности или необходимость дополнительного инвестирования.

- Отсутствие необходимого уровня контроля и возможностей мониторинга, что зачастую приводит к срыву планов выполнения работ и превышению выделенных бюджетов. С помощью инструментальных средств менеджеры проектов принимают решения на основе данных, поступающих только в реальном времени, не учитывая возможные последствия в будущем.

- Неконтролируемые изменения, поступающие от заказчиков и потребителей в виде новых идей относительно разрабатываемого программного обеспечения. Изменения могут существенно влиять на ход и успех проекта, поэтому необходимо оценивать предлагаемые изменения и реализовывать только одобренные, контролируя этот процесс с помощью соответствующих программных инструментов.

- Невозможность гарантировать качество и надежность программ из-за отсутствия гарантий отсутствия ошибок в программах вплоть до формальной сдачи заказчиком.

Все вышеперечисленные проблемы могут являться следствием недостаточной зрелости установленных в проекте процессов, например в соответствии с упомянутой ранее моделью СММІ [1]. При этом решение проблемы подходов и возможностей построения проектных прогнозов будет являться одним из методов развития и поддержания процессной зрелости.

3. Зрелость процессов разработки

На сегодняшний момент существует множество разнообразных методологий построения процесса разработки ПО, и у каждой из них есть свои достоинства и недостатки, области применения, в которых определенные из них наиболее эффективны. Все эти методологии преследуют целью улучшение производственного процесса, который позволил бы наиболее эффективно и качественно производить программные продукты [6]. Эффективность протекающих процессов описывается широким понятием зрелости, которое включает в себя, в частности, следующие аспекты:

- наличие возможностей по управлению процессами разработки, а также количественной основы для принятия решений;
- четкое распределение и понимание участниками сфер ответственности;

- использование пригодных и оптимальных способов работ, их четкое планирование;
- возможность совершенствовать существующие процессы, основываясь на качественных и количественных характеристиках;
- возможность наблюдать за качеством программного продукта и удовлетворенностью заказчика.

Одной из наиболее популярных, востребованных и весомых методик является модель построения и оценки зрелости процессов разработки программного обеспечения CMMI-DEV (Capability Maturity Model Integration for Development) [1]. Данная модель определяет 5 уровней качества производственного процесса в организации, каждый из которых служит фундаментом для перехода на следующий уровень:

- Начальный уровень (производственный процесс характеризуется как создаваемый каждый раз под конкретный проект, а иногда даже как хаотический. Определены лишь некоторые процессы и успех проекта зависит от усилий индивидуумов).

- Повторяемый уровень (установлены основные процессы управления проектом, позволяющие отслеживать затраты, следить за графиком работ и функциональностью создаваемого программного решения, установлена дисциплина процесса, необходимая для повторения достигнутых ранее успехов в проектах разработки подобных приложений).

- Определенный уровень (производственный процесс документирован и стандартизован как для управленческих работ, так и для проектирования, интегрирован в стандартный производственный процесс организации, во всех проектах используется утвержденная адаптированная версия стандартного производственного процесса организации).

- Управляемый уровень (собираются подробные количественные показатели производственного процесса и качества создаваемого продукта, производственный процесс и продукты оцениваются и контролируются с количественной точки зрения).

- Оптимизирующий уровень (постоянное совершенствование процесса достигается благодаря количественной обратной связи с процессом и реализации передовых идей и технологий).

CMMI-DEV описывает набор процессных практик, разделенных по процессным областям. Для перехода на некоторый уровень производственный процесс должен содержать минимальный необходимый набор практик, а его участники – использовать методы и инструменты для их реализации и поддержания. Понятие прогнозирования и способ получения проектных прогнозов в аспекте CMMI может трактоваться как метод совершенствования процесса, базирующийся на практиках 4-го уровня и дающий возможность организации соответствовать 5-му уровню зрелости.

4. Существующие решения и тенденции

На сегодняшний момент прогнозирование широко исследуется и применяется практически во всех областях человеческой деятельности; проектное управление, в том числе в сфере разработки программного обеспечения, не является исключением. В общем случае под управлением проектами подразумевается область деятельности, в ходе которой определяются и достигаются определенные цели, а также оптимизируется использование ресурсов (таких как время, деньги, трудозатраты и др.) в рамках некоторого проекта (определяющего конечный результат и ресурсные ограничения).

Многие крупные компании – поставщики программных систем управления продуктами, проектами и/или портфелями проектов – используют соответствующие формулировки при позиционировании и рекламе своих продуктов. Так, например, IBM Rational, заявляет, что ее решения помогают компаниям согласовать инвестиции в программное обеспечение и продукты с бизнес-целями и повысить прогнозируемость успеха проектов за счет применения лучших практических методов работы и измерения показателей в сопоставлении с ними [7]. Ее решения для управления продуктами повышают прогнозируемость их успеха благодаря более эффективному принятию решений, для управления проектами – оптимизируют реализацию, а для управления портфелями – позволяют согласовывать инвестиции с целями, сохраняя нацеленность на результат.

IBM Rational, один из крупнейших представителей в своей индустрии, взят в качестве примера, но список компаний, которые хотели бы использовать термин «прогнозируемость» в описании своих продуктов для управления программными проектами, огромен. К нему относятся все производители решений для сквозного моделирования и проектирования архитектуры программных продуктов (IBM Rational, Sparx Systems, Borland Inc. и др.), систем контроля дефектов и заявок на изменение (Atlassian, Mozilla Foundation, Borland Inc., Microsoft и др.), средств планирования и управления проектами (Microsoft, Assembla, Atlassian, eGroupWare и др.).

При этом важно отметить, что подавляющее большинство подобных типовых коробочных решений не решают задачу прогнозирования, а лишь выполняют часть необходимых условий, требующихся для формирования прогнозов (или, говоря в терминах СММИ, реализуют ряд необходимых процессных практик). Они могут предоставлять только инструмент управления информацией (например, решения для планирования проектов) или средства агрегации необходимых данных по некоторой области проектной деятельности (например, контроль дефектов и/или изменений в программном продукте). В целом такие системы и средства позволяют решать одну из задач прогнозирования – сбор необходимой информации для последующего формирования прогнозов.

5. Цели и задачи исследования

Представленная информация о предметной области позволяет нам подойти к определению цели проводимого исследования, которая заключается в формировании подхода, позволяющего совмещать понятия прогнозирования и управления проектами по разработке ПО. Возможность формирования и анализа научных гипотез в данной области позволит частично или полностью нивелировать существующие проблемы, обеспечит более эффективное принятие управленческих и технических решений. С точки зрения модели СММИ-DEV мы также можем говорить о методе повышения зрелости процессов разработки, который позволит проекту, при выполнении прочих необходимых условий, соответствовать 5-му уровню СММИ [1].

При этом одной из важнейших задач является формирование унифицированного подхода к реализации прогнозирования в проектах разработки, что позволит применять его вне зависимости от условий ведения проекта, выбранной методологии, размера команды, типа разрабатываемого продукта и других качественных и количественных характеристик.

В рамках данной статьи рассматривается ряд базовых принципов, позволяющих сформировать пути достижения поставленной цели. В частности, вводится понятие проектной модели, определяются базовые принципы анализа, а также способы формирования прогнозов с использованием математического моделирования.

Проектная модель и прогнозирование

В настоящее время моделирование часто применяется при решении разных задач разработки компьютерных систем: от анализа нагрузки частного программного решения (например, модель нагрузки) до изучения различных технологических и бизнес-процессов (модель процесса). Понятие модели в среде управления программным проектом облекает в конкретную форму общую идею моделирования применительно к процессу разработки ПО. В этом смысле проектная модель выступает в виде абстракции, представляющей различную информацию о проекте, а также средства управления этой информацией.

1. Значение модели в разработке программного обеспечения

Будучи связанной с реальным проектом, его модель может быть использована для реализации работы с такими активностями, как управление информацией о текущем состоянии проекта, управление распределением работ, наблюдение за ходом выполнения работ и оценка связанных с этим показателей, мониторинг показателей качества, вычисление метрик, подготовка и генерация отчетов и многое другое [8]. Такой подход позволяет отойти от сиюминутного реагирования на возникающие проблемы и дает возможность непрерывно наблюдать и анализировать состояние проекта за счет сбора информации, связанной с различными сторонами процесса проектирования, оценки различных параметров, а также планировать решения на основе полученных оценок [9].

Моделирование такого рода позволяет приблизиться к решению задачи прогнозирования проектных характеристик (например, вероятности успешного завершения работы или предполагаемого уровня качества кода после завершения фазы тестирования).

2. Прогнозирование и примеры моделей

Сегодня исследования систем прогнозирования проектных характеристик в области разработки программных продуктов считаются важными и перспективными. Они становятся возможными благодаря появлению данных и средств, необходимых для решения подобного рода задач. Многие из них оперируют понятиями «фактор» или «индикатор», на основе которых делаются те или иные прогнозы. Существующих формальных методик может быть недостаточно для обеспечения необходимого качества и точности результатов, поэтому нередко прибегают к имитационному моделированию, которое, по мнению некоторых исследователей, позволяет решить эту проблему [10].

Несмотря на наличие определенных результатов в данной области, практически все описанные методы решают частные задачи прогнозирования, используя существующий ограниченный набор

математических подходов. В большинстве своем они фокусируются на прогнозировании в двух основных областях, а именно:

- трудозатраты на создание и разработку продуктов в заданное время [11], в том числе оценки конечной стоимости, что соответствует классической модели трехстороннего ограничения (время, стоимость, функциональность), которое лежит в основе современного управления проектами [12];
- качество конечного продукта, в том числе количество дефектов на единицу исходного кода или другие характеристики, например стабильность работы системы.

Методы прогнозирования пытаются применять, чтобы ответить на вопрос о стоимости продукта и необходимых инвестициях (что становится возможным благодаря накоплению исторической информации о завершенных проектах), а также для получения представления о качестве результата, в который собственно инвестиции и вкладывают. До сих пор подобные решения не нашли воплощения в стандартных инструментальных средствах управления программными проектами, а зачастую просто описаны в научных изданиях либо частным образом используются в компаниях-производителях программного обеспечения.

Что касается областей, на которых сегодня фокусируются исследователи, то среди них можно выделить:

- собственно методы прогнозирования частных проектных характеристик [13–15];
- более эффективное использование ограниченного числа математических аппаратов, в том числе их комбинаций для получения лучших результатов [16];
- исследование способов обучения моделей для увеличения точности прогнозов.

Далее приведены некоторые примеры исследований в области прогнозирования частных проектных характеристик.

Исследование, проведенное в августе 2009 г. группой ученых под руководством Томаса Циммермана (Thomas Zimmerman) [17], было направлено на изучение кросспроектных прогнозов качества, а именно на анализ использования исторических данных предшествующих завершенных проектов внутри и вне компании. Необходимость исследования была обоснована недостаточностью собственных входных данных в новых проектах для прогнозирования качества разрабатываемых продуктов, что наводило на мысль о возможности переиспользования данных от сторонних компаний. Результаты исследования, в рамках которого было проведено более 600 кросспроектных прогнозов, говорят о практической возможности использования такого подхода, но и о необходимости предельной аккуратности в выборе ключевых факторов для анализа, чтобы избежать большой погрешности результатов.

Исследование, проведенное Мартином Шепардом (Martin Shepperd) [18] в рамках прогнозирования трудозатрат на создание программного продукта с помощью аналогии (Case Based Reasoning – CBR) в 2001 г., направлено на выявление типовых проблем в использовании метода. Мотивом к проведению исследования были результаты работы нескольких практикующих команд, свидетельствующие о совершенно разных по качеству результатах. Основными проблемами были признаны недостаточная подготовка входных данных (изменения которых критически сказывались на результатах, что говорит о малой стабильности системы), а также различные варианты параметризации самого метода оценки данных и формирования прогноза.

Достаточно интересным исследованием, находящимся в одной из двух основных областей (трехстороннее ограничение), является работа А. Демело (A. Demelo) и А. Санчес (A. Sanchez) в области прогнозирования задержек в проектах по поддержке программных продуктов с использованием байесовских сетей (Bayesian Networks) [19]. Основа исследования лежит в проблеме оценки времени и затрат, необходимых для такого рода проектов из-за большого количества неопределенностей в его процессах. Причиной этому даже в хорошо описанных и налаженных проектах являются приходящие заявки на изменения, которые могут иметь достаточно большое влияние на продукт или систему. Основой для прогнозирования временных задержек является опыт соответствующих специалистов, используемый в качестве входных данных.

3. Перспективы задач прогнозирования

Исследования в области прогнозирования проектных характеристик являются одним из наиболее перспективных направлений в области управления разработкой программного обеспечения, и это обусловлено следующими причинами:

- На текущий момент точные прогнозы в области разработки ПО востребованы ввиду больших инвестиционных вложений в ее рост и развитие; необходимо иметь возможность качественно оце-

нить необходимые капиталовложения в разработку программного продукта и сформировать ожидания от будущего результата.

- Существующие методы прогнозирования решают в первую очередь частные задачи, в том числе имеют predetermined фиксированный метод анализа и набор входных данных, недостаточную гибкость. Кроме этого, практически все методы сосредоточены в двух основных областях – трехстороннее ограничение проекта и качество результата.

- Решения, позволяющие формировать прогнозы характеристик программных проектов, имеют непосредственное отношение к современной и активно развивающейся области систем помощи принятия решений.

Проектное моделирование

Понятие системы происходит от древнегреческого слова *συστήμα*, означающего «сочетание», и представляет собой множество взаимосвязанных объектов и ресурсов, объединенных в единое целое и, возможно, противопоставляемое среде или суперсистеме. Системы как некоторого рода абстракции над реальными объектами или группами объектов обладают большим количеством свойств, связанных с целями, функциями, структурой, ресурсами и способами взаимодействия с окружающей средой. Данные свойства, в частности, позволяют рассматривать любой неэлементарный объект как подсистему целого (к которому рассматриваемый объект относится), выделив в нём отдельные части и определив взаимодействия этих частей, служащих какой-либо функции.

1. Проект разработки как система

Предполагая, что проект разработки не является элементарным объектом, мы можем применить к нему системную абстракцию, воспользоваться некоторыми свойствами систем (в частности, альтернативности развития) и возможностями проведения исследований для решения задач прогнозирования жизненного цикла и параметров программного проекта.

Так, например, одной из основных методологий описания на метаязыке и последующего исполнения (моделирования во времени) бизнес-процесса, которым, в частности, является проект разработки, служит BPM (Business Process Modelling – формальный язык описания бизнес-процессов) [20]. Стандарт предлагает специальную нотацию для описания процессов – BPMN (Business Process Modelling Notation), а также теоретические возможности последующего моделирования [21].

2. Структура проектной системы

Обладая свойством структурности, проектная система состоит из большого числа в общем случае также неэлементарных объектов (или подсистем), которые взаимодействуют между собой для достижения синергии в решении поставленных задач, а также ряда процессов и законов, по которым это взаимодействие происходит.

Типовые объекты, входящие в подобную систему, включают:

- участников проекта – проектную команду – сотрудников, имеющих определенную роль (например, разработчик, руководитель проекта, архитектор);
- создаваемые в результате работы артефакты (исходный код, документация, программный продукт);

- совокупность рабочего окружения, включая аппаратную и программную составляющие.

Взаимодействие этих объектов в системе происходит благодаря протекающим в ней процессам (рис. 1), которые определяются рядом факторов, в частности:

- принятой методологией разработки ПО (например, водопадная или спиральная модель);
- установленным процессом ведения проекта и создания артефактов;
- необходимостью взаимодействовать с суперсистемой (т.е. окружением, в котором выполняется проект).

Процессы, в свою очередь, характеризуются описывающими их параметрами, а также логической взаимосвязью с другими объектами системы. Примером процесса может являться разработка программного модуля, а связанные с ней объекты включать соответственно разработчиков и программный код.

3. Правила и факторы взаимодействия

Для понимания качества происходящих внутри проектной системы процессов и изменений необходимо ввести дополнительные составляющие модели, которые, в свою очередь, должны помочь ответить, в том числе, на следующие вопросы:

- Как описать состояние системы с учетом происходящих в ней процессов?

- Как охарактеризовать происходящие в системе процессы и их влияние на всю систему?
- Как определить степень необходимого влияния для «качественного» скачка в эволюции системы?
- Что может одновременно определять и корректировать происходящий производственный процесс и служить элементом его обратной связи?

Таковыми дополнительными составляющими являются правила и факторы.

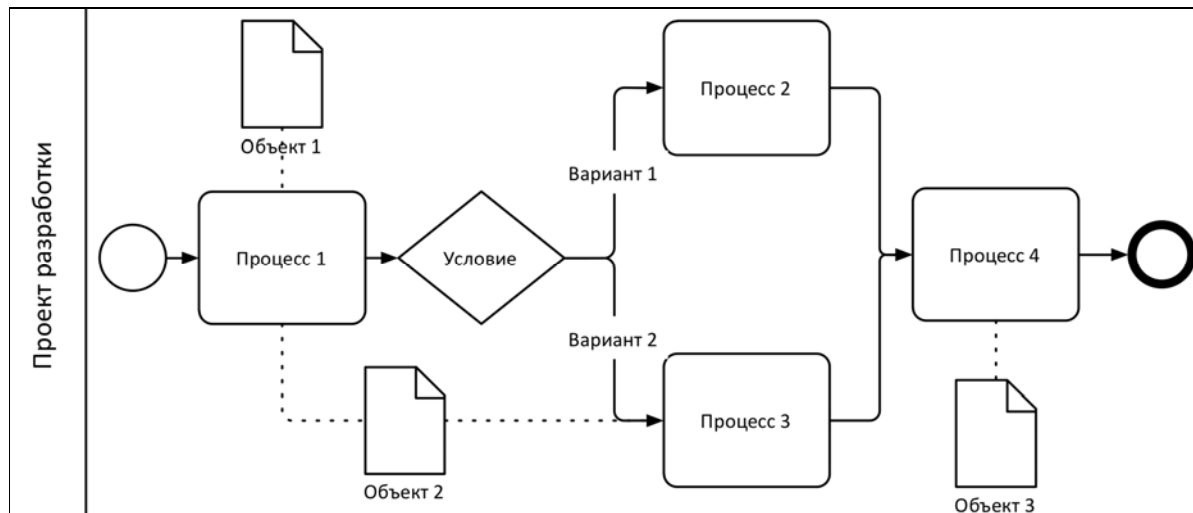


Рис. 1. Пример протекающих в проекте процессов в нотации BPMN

Правила могут описать границы выполнения процессов и жизни системы, определить исключительные и крайние (в том числе ошибочные) ситуации, задать условия переходов системы из состояния в состояние и определить логику взаимодействия протекающих внутри процессов. Так, например, на рис. 1 одним из правил процесса 1 может являться необходимость проверки наличия объекта 1 как обязательного входного условия.

Факторы в отличие от правил являются более сложными, двойственными составляющими. С одной стороны, фактор – это причина, движущая сила какого-либо процесса, определяющая его характер или отдельные его черты, которую можно изменять и тем самым управлять процессом; с другой – показатель (например, численный), описывающий состояние процесса и частично всей проектной системы. Примером фактора на рис. 1 для процесса 1 может являться размер объекта 1.

Введя понятия правила и фактора в проектную систему и среду, в которой она находится, мы можем определить внутренние процессы и описать взаимодействия, происходящие как в системе, так и между системой и внешним миром. Следует также отметить, что введенные сущности становятся атрибутами как системы в целом, так и протекающих в ней процессов (рис. 2). Конечный необходимый набор правил и факторов, необходимых для моделирования проектной системы, – лишь вопрос предположения и упрощения, проводимых при исследовании любого сложного реального объекта с помощью системной абстракции.

4. Принципы моделирования

Предлагаемый подход к моделированию проектной системы содержит в себе 3 основополагающих этапа (рис. 3):

- Описание структуры проектной системы, которое можно выполнить с помощью одной из существующих нотаций (например, BPMN) либо их расширений. Применение стандартных средств позволит также использовать существующие средства визуализации бизнес-процессов.
- Добавление контекста, а именно параметров, правил и факторов, выбор которых определяет решаемые при моделировании задачи анализа и прогнозирования, а также персонализирует протекающие в системе процессы. Полученная в результате структурного и контекстного описания модель может быть сохранена в некотором формате, например в XPD (XML Process Description Language – язык описания процессов) с необходимыми расширениями [22].
- Моделирование проектной системы, которое позволит проиграть протекающие процессы и взаимодействия во времени, проанализировать их последствия и получить необходимые значения спрогнозированных параметров и характеристик.

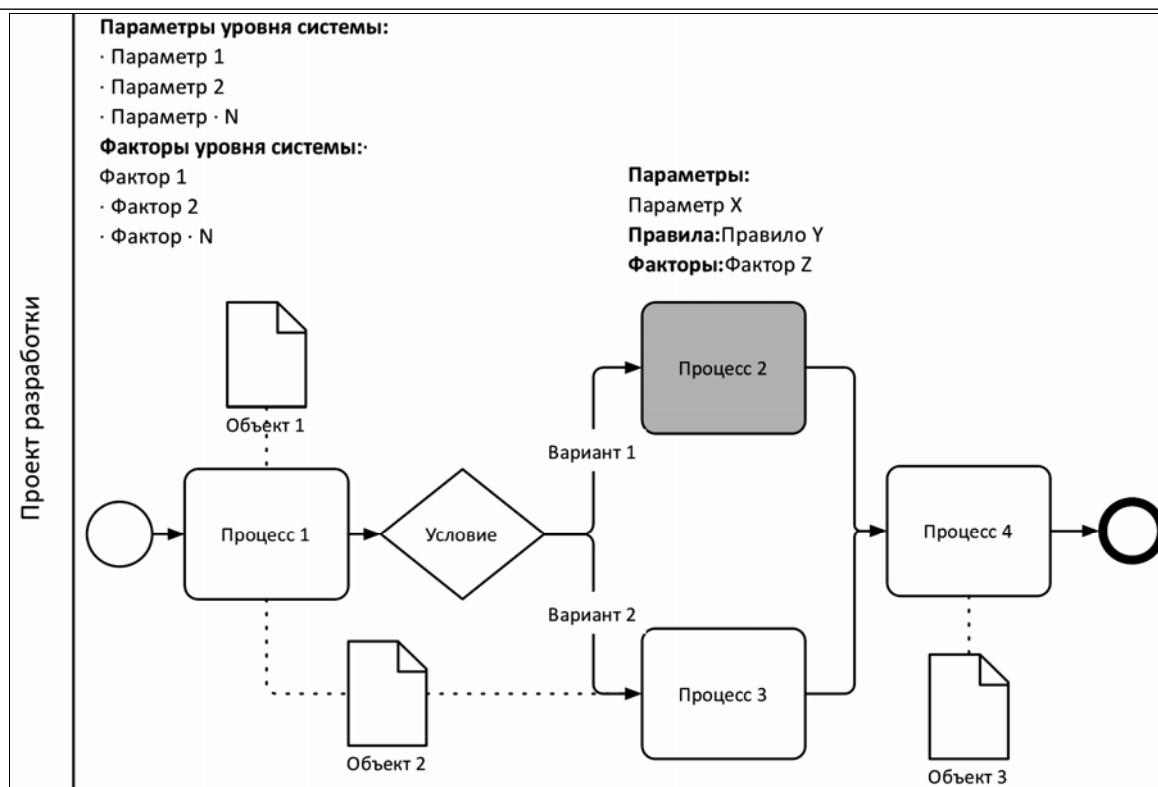


Рис. 2. Атрибуты модели и протекающих в ней процессов (расширение BPMN)

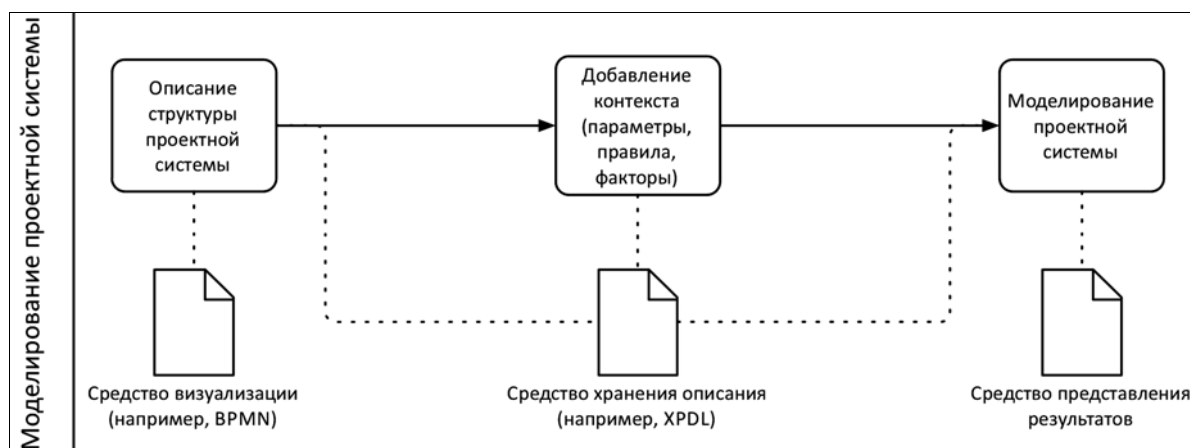


Рис. 3. Этапы моделирования проектной системы

В качестве базового подхода исследования проектной системы и прогнозирования ее характеристик разумно использовать дискретно-событийное имитационное моделирование, так как оно позволяет:

- описывать работу проектной системы, для которой построение исключительно аналитической модели не представляется практически возможным;
- имитировать поведение системы во времени, ускорять или замедлять его при необходимости;
- абстрагироваться от непрерывной природы событий, происходящих в системе, и базироваться только на основных происходящих в ней событиях.

Использование дискретно-событийного моделирования позволит исследовать только основные события или фазы жизни выбранной проектной системы, т.е. только то, что представляет для нас интерес. В частности, такой подход позволит исследовать проекты различной сложности, а также моделировать проекты разного типа, что соответствует одной из задач исследования. Кроме этого, имея универсальные элементы для описания проектов, будет возможно получать модели с различной детализацией (от самых простых – всего несколько фаз жизненного цикла, до более детальных, где каждая крупная фаза может содержать несколько вложенных этапов).

Таким образом, сам процесс моделирования протекает в дискретном системном времени в соответствии с описанной проектной структурой, параметрами и введенными правилами. Факторы взаимодействия при этом являются базовыми анализируемыми величинами, которые с одной стороны, изменяются в процессе жизни системы по некоторым предопределенным законам, а с другой – влияют на состояние и развитие системы в целом. Задание начальных значений факторов и параметров является задачей обучения системы, которое может быть выполнено, например, с использованием исторической информации, накопленной в результате ряда уже завершённых программных проектов.

Дискретное моделирование позволяет свободно перемещаться в системном времени в любом направлении, проигрывая протекающие взаимодействия и в том числе получая необходимые прогнозы проектных характеристик. Применение вспомогательных математических аппаратов (в частности, интерполяции и экстраполяции) дает возможность получать значения необходимых характеристик между существующими дискретными значениями, а также вне определенного временного интервала.

Дальнейшие шаги исследования

Дальнейшее исследование и развитие предложенного подхода к анализу процесса управления проектами по разработке ПО, которое базируется на выделении системной абстракции, определении свойственных ей правил и факторов, и последующему моделированию, включает в себя:

- Определение, детализацию требований и выбор методов формального описания модели и внутреннего системного взаимодействия. В частности, необходимо определить способ формирования моделей проектов, нотацию для работы с правилами и факторами, предложить при необходимости систему их классификации.

- Анализ и предложение необходимых базовых методов моделирования с учетом выбранных нотаций и существующих математических аппаратов. При этом исключительно важно, чтобы выбранные методы были релевантны решаемой задаче и предполагали возможность формальной и практической верификации.

- Выбор существующего или разработку собственного программного средства анализа сформированной модели.

- Сбор данных для целей обучения системы и постановки различных экспериментов.

- Определение путей верификации корректности работы предложенного подхода, в том числе с привлечением экспертов в области разработки ПО.

Заключение

Совмещение прогнозирования и управления проектами на сегодняшний момент является одной из перспективных задач для исследований в области разработки программного обеспечения, что обусловливается желанием лучше понимать и контролировать инвестируемые бизнес-процессы. Существующие на текущий момент математические аппараты позволяют применять к программным проектам понятие системной абстракции, что открывает широкие возможности по их анализу. Так, любой проект разработки может быть представлен в виде некоторой модели, над которой можно совершенно безбоязненно проводить необходимые эксперименты (чего нельзя сделать с реальным программным проектом). Более того, если рассматривать предложенный подход в аспекте СММИ-DEV, то мы можем говорить о методе, позволяющем реализовать практики, соответствующие 5-му уровню зрелости организации.

Определение подхода (метамодели) к описанию программных проектов, протекающих в них процессов и взаимодействий, а также способа моделирования позволяет:

- анализировать необходимые качественные и количественные характеристики в разное время жизни проекта;

- прогнозировать значения проектных характеристик в будущем;

- иметь обратную связь относительно эффективности выстроенных и планируемых к внедрению процессов и процессных практик;

- иметь качественную основу для принятия тех или иных управленческих или технических решений;

- использовать проектную модель как систему помощи принятия решений.

Предложенная метамодель программных проектов, сочетающая в себе этапы структурного описания и наложения специфического контекста (правил и факторов) вкупе с возможностью про-

дить моделирование, является достаточно гибкой и подходит для описания разнородных проектов, т.е. позволяет решать общую задачу анализа и прогнозирования в области управления проектами.

Литература

1. CMMI Product Team CMMI for Development, Version 1.3 // Software Engineering Institute. – Pittsburgh, Pennsylvania: Carnegie Mellon University, 2010 [Электронный ресурс]. – URL: <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm> (дата обращения: 25.12.2011).
2. Арефьева Н.Т. Прогнозирование и его социокультурные цели // Знание. Понимание. Умение: электронный журнал. – 2010. – №4 [Электронный ресурс]. – URL: <http://www.zpu-journal.ru/e-zpu/2010/4/Arefieva/> (дата обращения: 15.11.2011).
3. Соммервилл И. Инженерия программного обеспечения (Software Engineering). – 6-е изд. – М.: Вильямс, 2002. – 642 с.
4. Beecham S. Software Process Improvement Problems in Twelve Software Companies / S. Beecham, T. Hall, A. Rainer // Empirical Software Engineering. – 2003. – Vol. 8. – P. 7-42.
5. Glass R. Facts and Fallacies of Software Engineering. – Addison-Wesley, 2008. – 224 p.
6. Модель зрелости процессов разработки программного обеспечения / М. Паулк, Б. Куртис, М. Хриссис, Ч. Вебер и др. – М.: Интерфейс-Пресс, 2009. – 402 с.
7. Управление продуктами, проектами и группами решений. – 2011 [Электронный ресурс]. – Дата обновления 20.11.2011. – URL: <http://www-01.ibm.com/software/ru/rational/ppm/> (дата обращения: 20.11.2011).
8. Atlee J.M. Modeling in Software Engineering // Proceedings of the 29th International Conference on Software Engineering. – Minneapolis, 2007. – 23 p.
9. Pyshkin E. Virtualization in the software management / E. Pyshkin, A. Bazhenov // Proceedings of 12th International Conference on Humans and Computers (HC2009). – Hamamatsu, Japan: Shizuoka University, 2009. – P. 86–88.
10. Shepperd M. Building Prediction Systems for Software Engineers / M. Shepperd, M. Cartwright, G. Kadoga // Empirical Software Engineering. – 2000. – Vol. 5, № 3. – P. 175–182.
11. MacDonell S. A Comparison of Modeling Techniques for Software Development Effort Prediction / S. MacDonell, A. Gray // Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems. – Dunedin, New Zealand, 1997. – P. 40.
12. Vaes K. Devil's Triangle of Project Management PM Hut. [Электронный ресурс]. – URL: <http://www.pmhut.com/devils-triangle-of-project-management> (дата обращения: 20.11.2011).
13. Pendharkar P.C. A probabilistic model for predicting software development effort / P.C. Pendharkar, G.H. Subramanian, J.A. Rodger // IEEE Transactions in Software Engineering. – 2005. – Vol. 31. – P. 615–624.
14. Fenton N. Probabilistic Model for Software Defect Prediction / N. Fenton, P. Kraus, M. Neil // IEEE Transactions in Software Engineering. – 2001. – Vol. 27. – 130 p.
15. Rosli M.M. Fault Prediction Model for Web Application Using Genetic Algorithm / M.M. Rosli, N. H. I. Teo, N. Shahida et al. // International Conference on Computer and Software Modeling. – Singapore, 2011. – P. 63–70.
16. MacDonell S. Combining Techniques to Optimize Efforts Predictions in Software Project Management / S. MacDonell, M. Shepperd // Systems and Software. – Elsevier Science Publishing Company, 2003. – P. 91–98.
17. Zimmermann T. Cross-project Defect Prediction / T. Zimmermann, N. Nagappan, H. Gall, E. Giger, Murphy B. // Proceedings of the 7th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE). – Amsterdam, Nederland, 2009. – P. 91–100.
18. Kadoda G. Issues on the Effective Use of CBR Technology for Software Project Prediction / G. Kadoda, M. Cartwright, M. Shepperd // Lecture Notes in Computer Science. – 2001. – Vol. 2080. – P. 276–290.
19. Demelo A. Software maintenance project delays prediction using Bayesian Networks Expert Systems with Applications / A. Demelo, A Sanchez // Expert Systems with Applications. – 2008. – Vol. 34. – P. 908–919.
20. Business Process Management Initiative. – 2011. [Электронный ресурс]. – URL: <http://www.bpmi.org> (дата обращения: 16.11.2011).

21. Havey M. Essential Business Process Modeling // O'Reilly Media Inc. – 2005. [Электронный ресурс]. – URL: <http://oreilly.com/catalog/9780596008437/preview#preview> (дата обращения: 02.12.2011).

22. Stephen A. XPDL and BPMN Workflow Handbook. – 2003 [Электронный ресурс]. – URL: <http://www.wfmc.org> (дата обращения: 01.12.2011).

Баженов Андрей Сергеевич

Аспирант каф. компьютерных систем и программных технологий Санкт-Петербургского политехнического университета (СПбГПУ)

Тел.: +7 (812) 297-22-38

Эл. почта: andrey.s.bazhenov@gmail.com

Ицыксон Владимир Михайлович

Доцент каф. компьютерных систем и программных технологий СПбГПУ

Тел.: +7 (812) 297-22-38

Эл. почта: vlad@ftk.spbstu.ru

Bazhenov A. V., Itsykson V.M.

Applying of forecasting in software design process

Processes forecasting is one of the major questions within the area of commercial software development. This aspect is reflected in different development methodologies, maturity models like CMMI, however it still lacks any kind of universal solution. The research being conducted is focusing on the modern trends within software development processes' multi-factor analysis. Besides that it formulates an approach for hypothesizing within software project management which should improve overall development process predictability.

Keywords: Software development, model, factor, forecasting, CMMI.
