

АНАЛИЗ ИНФОРМАЦИОННОЙ СТРУКТУРЫ ПРОЦЕССА ВЫПОЛНЕНИЯ ПРОЕКТА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С. А. Полицын

Московский авиационный институт (Национальный исследовательский университет)

Поступила в редакцию 03.03.2016 г.

Аннотация. Процесс выполнения проекта разработки программного обеспечения (ПО) динамичен, ошибка в планировании может привести к значительной задержке времени окончания проекта и/или существенному перерасходу ресурсов.

Для создания эффективных инструментов поддержки принятия решений при управлении проектом разработки ПО с возможностью прогнозирования хода его выполнения необходимо детально исследовать информационные процессы и структуры данных, которые в нем задействованы. Статья посвящена анализу информационной структуры процесса выполнения проекта разработки ПО.

Ключевые слова: управление проектами, система поддержки принятия решений, жизненный цикл ПО, информационная структура процесса разработки ПО.

Annotation. Wrong estimations of the software planning process often result in extra costs. It is necessary to analyze the involved informational processes and data structures of the software development process to create effective decision support systems with estimation functionality in the field of program engineering. The article is devoted to the informational structure analysis of a software development project.

Keywords: project management, resource allocation, decision support systems, project management system, software development project informational structure.

ВВЕДЕНИЕ

Интенсивное развитие информационных и коммуникационных технологий привело к формированию новых требований к планированию и управлению проектами в различных областях. Процесс выполнения проекта разработки программного обеспечения (ПО) динамичен, требования и набор задач могут часто меняться, ошибка в планировании может привести к значительной задержке времени окончания проекта и/или существенному перерасходу ресурсов. Одним из обязательных условий эффективной реализации проектов становится применение современных средств и инструментов управления проектами, основанных на использовании новых информационных технологий. Развитие специального программного обеспечения для планирования и управления проектами

обусловлено, в первую очередь, необходимостью максимальной интеграции наиболее эффективных методов, средств и инструментов теории управления проектами для оперативного доступа к аналитическим данным по состоянию проекта [1].

Независимо от подхода к построению процесса выполнения проекта разработки ПО («каскадного», итерационного, «гибкого»), точный расчет ресурсов, необходимых для реализации данного продукта при определенных требованиях к качеству, является одной из основных проблем в области управления проектами [2], поэтому существует потребность в разработке методов и средств, позволяющих руководителю оценить требуемые временные и человеческие ресурсы на основе всех имеющихся характеристик проекта: истории предыдущих подобных проектов, опыта и производительности сотрудников, специфики компании и т. п.

Существующие системы ведения проектов (MS Project, Primavera Project Planner, Spider Project и т. д.) предоставляют средства для описания работ проекта, связей между работами и их временных характеристик, построение сетевого графика проекта, поддержки информации о ресурсах и затратах по проекту и назначения ресурсов и затрат отдельным работам проекта, контроля над ходом выполнения проекта и использованием ресурсов, включающие инструменты отслеживания состояния задач проекта, процент завершения, подробные графические средства представления структуры, визуального построения диаграмм, средства составления отчетов, но не имеют достаточных средств для прогнозирования дальнейшего хода выполнения проекта и оценки вероятности его успешного завершения [3–5].

Для создания инструментов поддержки принятия решений при управлении проектом разработки ПО, обладающих возможностью прогнозирования хода его выполнения с учетом заданных ресурсов и определения вероятности успешного завершения проекта необходимо детально исследовать информационные процессы и структуры данных, которые в нем задействованы.

ПРОЦЕСС ВЫПОЛНЕНИЯ ПРОЕКТА РАЗРАБОТКИ ПО

Различные методики ведения проекта в общем случае отличаются принципами организации непосредственно процесса разработки ПО, но в любом случае, в том или ином виде обязательно присутствуют следующие этапы:

- инициация проекта и формулировка его цели;
- формирование требований для достижения поставленной цели;
- формирование списка задач проекта на основе требований;
- формирование команды (выделение ресурсов на проект);
- составление детального плана задач;
- выполнение задач и тестирование;

- анализ результатов проекта (или его итерации) и сопровождение.

На разных этапах проекта разработки ПО необходимо обрабатывать различную по своему составу, степени упорядоченности и структуре информацию. В зависимости от вида информации, с которой приходится иметь дело этапы проекта были объединены в три фазы: инициация проекта и составление списка задач, построение плана итераций проекта, выполнение задач по проекту. После завершения всех трех фаз процесса обычно проводится анализ результатов. Модель процесса выполнения проекта разработки ПО представлена на IDEF0 диаграмме (рис. 1). Далее рассмотрены виды информации каждой фазы проекта.

Первая фаза включает в себя начальные этапы зарождения проекта и начальное формирование проектной команды, на которых обрабатывается разнородная, очень слабо поддающаяся формализации информация. В результате завершения этой фазы на выходе получается общий список задач проекта.

Первая фаза включает в себя следующие процессы:

1. Сбор требований и формирование набора задач по проекту

После формирования цели проекта и определения общих требований производится уточнение требований и постановки задач. Основной сложностью на этом этапе является разнородность и недостаточная степень формализации первичных требований к разрабатываемым системам. Такие требования обычно описывают разрабатываемую систему с точки зрения ее назначения без учета технических деталей. Кроме того, требования могут исходить из разных источников – лиц или организаций, заинтересованных в выполнении проекта. Ими могут быть заказчик(и) проекта, аналитики, изучающие рынок, стандартизирующие организации, юридический отдел и т. д.

В качестве источника новых требований к системе можно также рассматривать ошибки, обнаруживаемые на этапе тестирования, поскольку в этом случае задача или ее части должны быть выполнены заново. От всех

заинтересованных лиц с разной интенсивностью поступают разнородные, обладающие различными приоритетами требования к разрабатываемой системе. Определение итогового приоритета каждого требования при этом состоит в объединении приоритета источника и приоритета этого требования в рамках источника. В самом общем случае необходимо рассматривать бесконечное количество требований. На основе всех требований составляется предварительный общий список задач.

2. Построение графа задач

При планировании проекта обычно на основе предварительного общего списка задач с учетом их связей и приоритетов руководитель проекта составляет граф задач [7]. Список задач, полученных от источников задач, анализируется и преобразуется к виду графа, который соответствует списку задач по одному проекту с учетом связанности задач, подзадач и их приоритетов.

Этот процесс достаточно трудоемкий даже для проектов среднего размера. Количество задач, участвующих в оценке, может быть больше сотни; определив их взаимосвязи, зачастую бывает весьма затруднительно рассчитать критический путь вручную, автоматизация этого процесса позволяет ускорить анализ план-графика проекта, а также дает возможность моделировать возможные варианты хода течения проекта по методу Монте-Карло.

3. Формирование общего списка задач

После построения графа задач необходимо проанализировать его с точки зрения согласованности задач по их связям и приоритетам. Так, например, одна из подзадач может иметь более высокий приоритет, чем родительская или предшествующие ей, в таком случае приоритет задач необходимо пересмотреть.

Для формирования очереди задач проекта по составленному графу необходимо оценить время выполнения (трудоемкость) каждой задачи. Существуют различные методики оценки времени выполнения задач, для прогнозирования проектов самым важным моментом является не абсолютная величина

оценки, а относительные оценки задач, получаемые путем сравнения времени выполнения задач между собой.

Следующий шаг – декомпозиция их на более мелкие, удобные для решения задачи. Уровень декомпозиции зависит от многих факторов, но на практике удобно выбирать такой размер задачи, которая была бы законченной и могла бы быть полностью интегрирована в разрабатываемое ПО в отведенные сроки. Для дальнейшей оценки проекта целиком очень важно, чтобы такие задачи выполнялись за примерно одинаковые интервалы времени, т. е. для разных проектных задач уровень декомпозиции был одинаковым с точки зрения времени их выполнения.

После прохождения оценки и обработки получается размеченный по весам и упорядоченный по приоритетам граф задач проекта, на основе рекурсивного обхода сетевого графа задач в соответствии с приоритетами создается общий список задач проекта (англ. “backlog”) с сохранением информации о приоритетах и взаимосвязях. [8]

Главной задачей **второй фазы** является получение плана проекта и формирование прогноза его выполнения. В задачи этой фазы также входит оценка вероятности выполнения плана проекта с помощью одной из моделей и перестроение получившегося плана в случае неудовлетворительного прогноза. К основным процессам второй фазы относятся:

1. Составление плана проекта

Для составления плана проекта предварительно необходимо выбрать длину отдельных итераций проекта, при следовании водопадной модели любой проект состоит только из одной итерации. Как правило, итерации выбираются одинакового размера по времени, даже если команда разработки не придерживается гибких методологий разработки.

В простом случае, когда не требуется учитывать зависимости между проектами, имеется стабильная команда разработки и пр., план итераций проекта можно получить, распределяя задачи из общего списка задач проекта между членами команды разработки, пока не закончится доступное время итерации. Аналогично процесс повторяется для

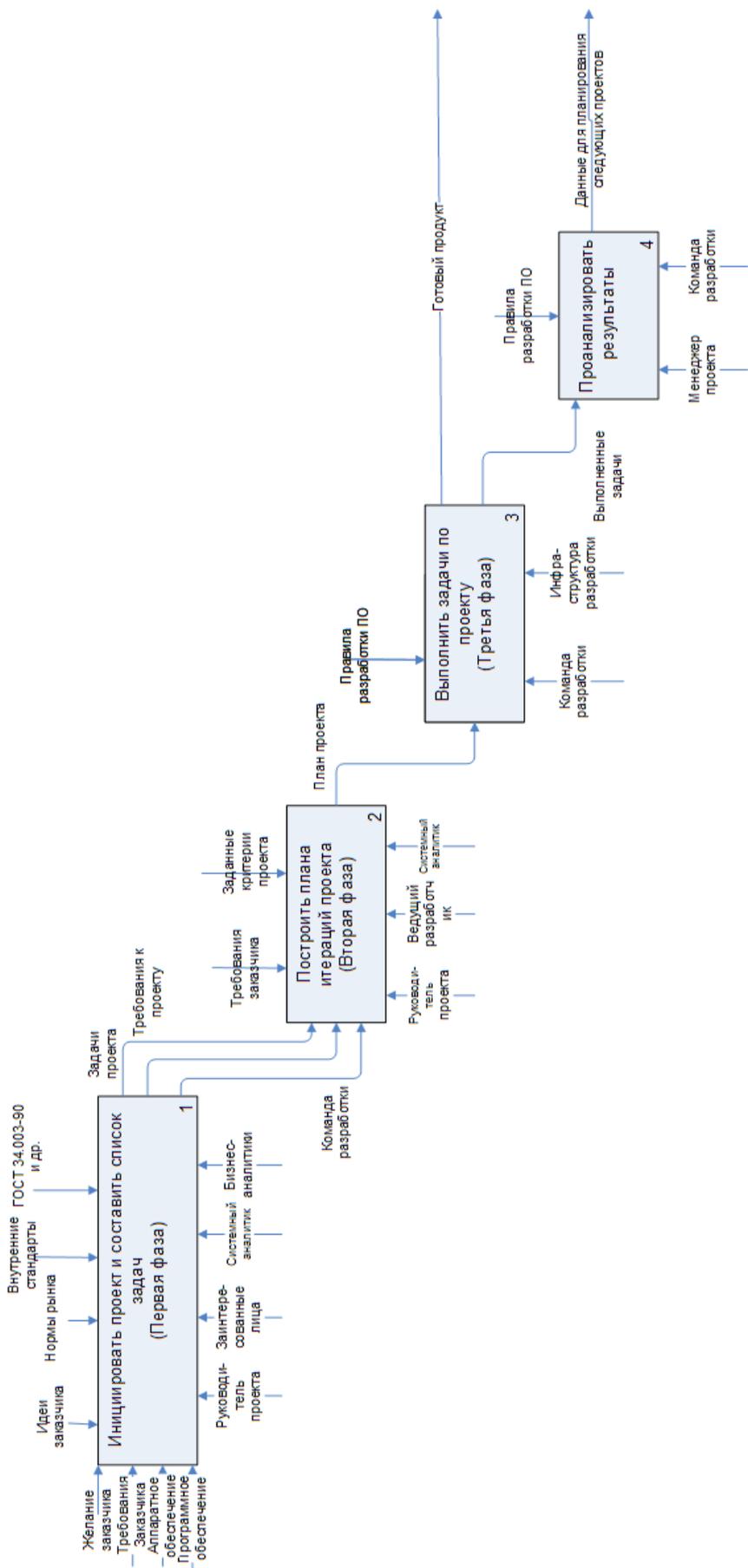


Рис. 1. Модель процесса выполнения проекта разработки ПО

остальных итераций, пока не будут распределены все задачи из списка. В результате получится расчетная дата окончания проекта и количество плановых итераций проекта.

2. Оценка плана проекта с помощью одной из моделей

Этот этап является одним из самых важных, однако во многих проектах он пропускается. Сроки, полученные на предыдущем этапе, являются грубым приближением. Для того, чтобы вычислить дату окончания проекта с более высокой точностью применяется какая-либо (или даже их сочетание) из методик предварительной оценки проекта. Ручное выполнение этой задачи – очень трудоемкий процесс, например, для метода Монте-Карло, ручные расчёты имитационной модели и всех ее «исходов» потребовали бы огромное количество времени. Поэтому для расчетов с помощью моделей применяются соответствующие программные средства.

3. Пересмотр плана проекта

Если в результате оценки с помощью какой-либо модели была получен неудовлетворительный прогноз успешного окончания проекта, то план проекта пересматривается и его построение повторяется.

Третья фаза включает в себя деятельность по программированию, тестированию, разработке дизайна и т.д. Эта фаза обычно занимает больше всего времени и на ней задействована вся команда разработки. В случае изменения состава входных данных может возникнуть необходимость возврата к предыдущим фазам.

ПРЕОБРАЗОВАНИЕ ДАННЫХ В ПРОЦЕССЕ ВЫПОЛНЕНИЯ ПРОЕКТА РАЗРАБОТКИ ПО

Преобразование требований источников к задачам для исполнения

Требования к ПО включают в себя **бизнес-требования** и **системные требования**. В первоначальном варианте бизнес-требования, поступающие от разных источников (рис. 2), обычно представлены в виде текстового описания, структурированного в виде списка, на основе анализа которого состав-

ляются системные требования или технического задания, для описания которых используется структурированный текст или UML диаграммы прецедентов в графической или текстовой форме.

После этого решается задача преобразования каждого требования в одну или несколько задач и расстановки их приоритетов. Одному требованию может соответствовать одна или несколько задач. На данном этапе **задача** включает в себя следующую информацию:

- *номер* – уникальный номер задачи в списке задач от источника;
- *название* – заголовок задачи, кратко характеризующий ее суть;
- *описание* – детальное описание задачи;
- *первоначальный приоритет* – приоритет задачи в списке задач от источника.

Для создания задач на основе требований необходим их тщательный анализ, привлечение различных специалистов для определения детальных требований и ограничений, определения приоритетов, поэтому автоматизация преобразования требований к спискам задач затруднительна. Кроме того, не все требования в проекте являются обязательными, так как на их реализацию всегда накладываются ограничения ресурсов: сроков и команды разработки, что также учитывается при определении приоритета задач.

Объединение списков задач

После описания требований и списков задач от всех источников составляется предварительный общий список задач по проекту, наибольшую сложность представляет собой определение приоритетов задач в нем на основе ранее заданных первоначальных приоритетов и приоритета источника.

На данном этапе происходит преобразование N списков задач к единому списку, в котором задача включает в себя следующую информацию:

- *номер* – уникальный номер задачи в общем списке задач
- *название* – заголовок задачи, кратко характеризующий ее суть;
- *описание* – детальное описание задачи;

- *приоритет* – приоритет задачи в общем списке задач.

Построение графа задач

После построения **предварительного общего списка задач** по проекту для каждой задачи определяются сложность и ее связь с другими задачами. Виды связей могут быть следующие:

1. Задача А зависит от задачи В, т.е. пока не выполнена задача В, невозможно начать выполнять задачу А.

2. Задача А является частью задачи В, т.е. пока не выполнена задача А, задача В не будет выполнена, при этом сложность задачи В равна 0.

3. Задача А ассоциативно связана с задачей В, т.е. необходимые изменения функционала программного продукта в задаче А связаны с ранее сделанными изменениями по задаче В.

Оценка сложности каждой задачи производится на основе экспертного мнения специалистов с учетом их знаний, опыта прошлых проектов и особенностей разрабатываемого продукта.

Для упрощения оценки сложности задачи разбиваются на более мелкие, в которых проще оценить объем необходимых изменений. Сложность задач обычно измеряется в человеко-часах, т. е. количестве часов работы одного специалиста.

Таким образом, на данном этапе описание задачи включает в себя:

- *номер* – уникальный номер задачи в общем списке задач;
- *название* – заголовок задачи, кратко характеризующий ее суть;
- *описание* – детальное описание задачи;
- *приоритет* – приоритет задачи в общем списке задач;
- *трудоемкость* – оценка сложности задачи в условных единицах, например, задачи в человеко-часах;
- *список зависимых задач* – список номеров задач, которые зависят от текущей;
- *список подзадач* – список номеров задач, которые являются частями текущей;
- *список связанных задач* – список ассоциативно связанных задач с текущей.

С учетом информации о зависимостях задач список задач преобразуется к **предварительному графу задач**. Информация о подзадачах и ассоциативно связанных задачах используется исключительно для упрощения работы и контроля процесса их выполнения.

Переопределение приоритетов задач

Построение общего списка задач для составления плана проекта затрудняется на данном этапе несогласованностью построенного графа задач по приоритетам, т.е., например, задача Z3 имеет первый приоритет, но зависит от задачи Z1 с третьим приоритетом, соответственно Z3 не может быть выполнена, пока не выполнена Z1, а в силу низкого приоритета Z1 будет выполнена не в первых итерациях проекта. Для решения этой проблемы переопределяются приоритеты задач, чтобы привести зависимые задачи к одинаковым приоритетам. Обычно повышается приоритет от задач, которые необходимы для решения других высокоприоритетных, т. е. в примере повышается до первого приоритет задачи Z1. После этого **предварительный граф задач** преобразуется с **графу задач** с согласованными приоритетами.

В общем случае задача согласования приоритетов решается путем их обсуждения руководителем проекта с заинтересованными лицами и командой разработки, при этом приоритеты могут быть переопределены в сторону понижения приоритета зависимой задачи. В описании задачи при этом изменяются только приоритеты, но могут добавляться новые задачи путем разбиения существующих на более мелкие с целью согласовать приоритеты, если для выполнения зависимой задачи достаточно решить часть предыдущей. В этом случае задачи добавляются в граф задач.

Построение общего списка задач

На основе графа задач строится общий список задач («backlog»), для дальнейшего построения **плана проекта** на данном этапе в описание задач добавляется *статус*, чтобы в последствии учитывать связи между задачами. В каждой команде разработки может

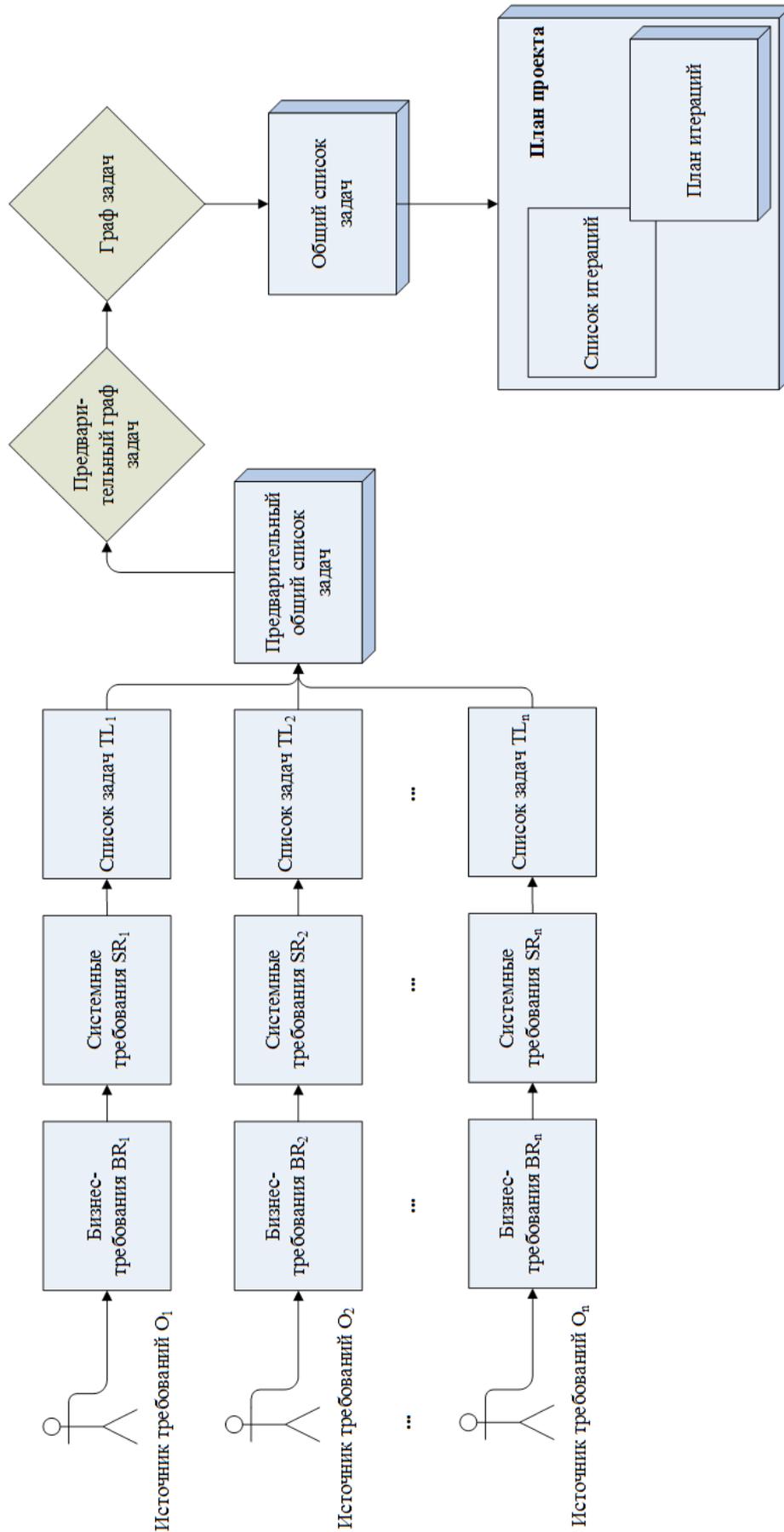


Рис. 2. Схема преобразования данных в процессе выполнения проекта разработки ПО

использоваться свой набор статусов задач в зависимости от построения процесса разработки и тестирования.

Построение плана проекта

План проекта включает в себя **список итераций** и **план итераций**. В зависимости от списка задач по проекту, его сроков и выделенных ресурсов определяется длина итерации и их количество. Для построения плана итераций задачи упорядочиваются по приоритетам, т. е. строится очередь задач с сохранением информации о связях между задачами и их статусе.

На каждую итерацию отбирается ряд задач в зависимости от их приоритета и трудоемкости, учитывая статус задач к данной итерации для формирования непротиворечивого плана итераций с точки зрения зависимостей между задачами.

План проекта является окончательными исходными данными для выполнения задач по нему. После каждой итерации он может изменяться с учетом статусов задач: если часть задач в итерации не была выполнена, то это требует пересмотра плана следующих за ней итераций с целью включения в них невыполненных задач и переноса зависимых от них задач.

В случае успешного проекта после завершения последней запланированной итерации все задачи, реализующие обязательные требования должны быть выполнены.

ПОДХОД К АВТОМАТИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ПО

Проведенный анализ показал, что процесс разработки ПО длительный, трудоемкий, характеризующийся большим числом преобразований данных на разных этапах проекта, особенно в вопросах, связанных с работой со списками задач по проекту в целом или по итерациям, так как именно эти данные отражают все вносимые в план проекта изменения, как со стороны заинтересованных лиц, так и стороны команды разработки. Для его упрощения применяется ряд средств автоматизации на различных фазах создания проекта.

На начальных этапах первой фазы, когда формируются требования и еще нет формализованного списка задач, автоматизацию процесса разработки составляют системы ведения проекта и системы хранения требований [3–5], позволяющие вносить, хранить, модифицировать, структурировать информацию о требованиях, часть из них поддерживает работу с формализованным представлением требований в форме UML-диаграмм прецедентов. Часто к этому добавляется возможность хранения многосвязных списков требований и задач. Полная автоматизация процесса формирования и формализации требований и создания задач на их основе сильно затруднена слишком большим влиянием всех участников проекта и прочих заинтересованных лиц, от которых поступают требования или ограничения, поэтому для процессов первой фазы используются только инструменты, упрощающие рутинную работу с данными по проекту.

С повышением уровня формализации передаваемой информации с одного этапа на другой во второй и третьей фазах с одной стороны появляется больше повторяющейся рутинной деятельности, которая может и должна быть автоматизирована, с другой стороны еще больше, чем в первой фазе, проявляется динамичность процесса разработки ПО и изменчивость требований. Поэтому повышается потребность в использовании средств автоматизации и создании новых средств поддержки принятия решений [6] для оперативного контроля за ходом выполнения проекта.

В результате проведенного анализа информационной структуры процесса разработки ПО были выявлены части процесса, которые нуждаются в автоматизации:

- *переопределение приоритетов задач и построение общего списка задач.*

Составление общего списка задач на основе предварительного графа происходит в два этапа: переопределение приоритетов задач и непосредственно обход графа для составления списка. Для решения обеих задач необходимо провести рекурсивный обход графа, при количестве вершин превышающем де-

сятки достаточно трудоемка для вычислений без применения ЭВМ, к тому же алгоритмы рекурсивного обхода достаточно хорошо изучены, поэтому автоматизация этого процесса должна проводиться в первую очередь.

- *автоматизация построения плана итераций и его проверки с помощью одной или нескольких моделей оценки.*

Как до начала, так и во время выполнения задач, план итераций может претерпевать множество изменений, это связано с изменчивостью требований к ПО и вероятностным характером процесса их выполнения в последствии, поэтому необходимо использовать существующие средства моделирования для проверки построенного плана проекта с точки зрения оценки успешности выполнения проекта и создавать новые для увеличения точности и упрощения получения оценок, что может достигаться путем автоматизации построения и использования этих моделей.

- *автоматизированный контроль хода выполнения задач и внесения корректировок в план проекта.*

В ходе выполнения задач большое значение имеет своевременность реагирования на изменения, для чего необходимо быстро получать актуальные данные по текущему состоянию задач, анализировать их с целью определения последствий и вносить корректировки в план проекта. Для решения этой проблемы необходимо создания новых средств контроля хода выполнения проекта и систем поддержки принятия решений по внесению изменений в план проекта.

ЗАКЛЮЧЕНИЕ

Проведенный анализ информационной структуры процесса разработки ПО показал, что в нем задействовано большое количество участников проекта и заинтересованных лиц, влияющих на ход его выполнения на всех этапах, при этом кроме непосредственно выполнения задач по реализации функционала ПО существует потребность в преобразовании большого количества данных, которые отражают динамичность процесса. В результате проведенного анализа информационной

структуры процесса разработки ПО были выявлены части процесса, которые нуждаются в автоматизации: переопределение приоритетов задач и построение общего списка задач, построения плана итераций и его проверки с помощью одной или нескольких моделей оценки, контроль хода выполнения задач и внесения корректировок в план проект. Создание систем поддержки принятия решений с такими возможностями даст руководителю проекта возможность быстрого получения оперативных данных для принятия решения и упростит этот процесс.

СПИСОК ЛИТЕРАТУРЫ

1. *Ройс У.* Управление проектами по созданию программного обеспечения / У. Ройс. – М. : Лори, 2007. – 424 с.
2. *Полицын С. А.* Математическая модель процесса выполнения задач при разработке программного обеспечения / С. А. Полицын, В. В. Шилов // Информационные технологии. – 2013. – № 10. – С. 24–29.
3. Сайт компании «Технологии управления Спайдер»: – Режим доступа: <http://www.spiderproject.com> – 03.03.2016.
4. Сайт компании Microsoft. Microsoft Project: – Режим доступа: <http://www.microsoft.com/project> – 03.03.2016.
5. Сайт компании Oracle. Primavera Enterprise Project Portfolio Management: – Режим доступа <http://www.primavera.com> – 03.03.2016.
6. *Ларичев О. И.* Системы поддержки принятия решений. Современное состояние и перспективы их развития / О. И. Ларичев, А. Б. Петровский // Итоги науки и техники. Серия: Техническая кибернетика. – Т. 21. – М. : ВИНТИ, 1987. – С. 131–164.
7. Project Management Institute, A Guide to the Project Management Body of Knowledge (5th ed.), 2013, Project Management Institute.
8. *Полицын С. А.* Система планирования проектов разработки программного обеспечения / С. А. Полицын // Вестник Воронеж. гос. ун-та. Сер. Системный анализ и информационные технологии. – 2013. – № 1. – С. 75–80.

С. А. Полицын

Полицын Сергей Александрович – ст. преподаватель, институт № 4, кафедра «Проектирование вычислительных комплексов», Московский авиационный институт (Национальный исследовательский университет).
Тел.: 8(495) 509-92-19
E-mail: pul_forever@mail.ru

Politsyn Sergey A. – senior lecturer, department of «Design of computing systems», Moscow Aviation Institute (National Research)
Tel.: 8(495) 509-92-19
E-mail: pul_forever@mail.ru