

УДК 004.4'22  
DOI: 10.36979/1694-500X-2023-23-12-88-95

## UML КАК ИНСТРУМЕНТ УПРАВЛЕНИЯ ПРОГРАММНЫМИ ПРОЕКТАМИ

*С.Ц. Манжикова*

*Аннотация.* Предлагается использование средства языка визуального моделирования систем UML для планирования самого процесса разработки сложного программного обеспечения с соблюдением условий и требований технологии, известной как Rational Unified Process – рационального унифицированного процесса. Анализ релевантных публикаций показал, что в сфере организации управления процессами разработки программного обеспечения используются самые разные способы визуального представления структуры и динамики таких процессов. Так, в ряде работ применяются для этого даже блок-схемы алгоритмов, которые лишь дополняются некоторыми комментариями о требуемой квалификации исполнителей, выполняющих работы в соответствии с той или иной выбранной технологией/методологией разработки программных систем. Разработка современного, а значит, сложного программного обеспечения обязательно должна начинаться с его предварительного проектирования с применением объектно-ориентированного метода, который базируется на языке UML. Поэтому представляется логически оправданным применение инструментов этого языка и для построения процессов управления программными проектами. В статье формулируются правила адаптации инструментов UML для моделирования процессов управления разработкой программ и дается пример использования UML диаграмм переходов состояний и деятельности для представления дисциплин, фаз, контрольных точек технологии Rational Unified Process.

*Ключевые слова:* унифицированный язык визуального моделирования систем; управление программными проектами; технология рационального унифицированного процесса разработки программных систем; фазы-состояния; дисциплины-подсостояния; контрольная точка milestone.

---

## UML ПРОГРАММАЛЫК ДОЛБООРЛОРДУ БАШКАРУУ КУРАЛЫ КАТАРЫ

*С.Ц. Манжикова*

*Аннотация.* Макалада Rational Unified Process – рационалдуу бирдиктүү процесс деп аталган технологиянын шарттарын жана талаптарын сактоо менен татаал программалык камсыздоону иштеп чыгуу процессин пландаштыруу үчүн UML системасынын визуалдык моделдөө тилинин каражатын колдонуу сунушталат. Тиешелүү басылмаларга талдоо жүргүзүү программалык камсыздоону иштеп чыгуу процесстерин башкарууну уюштуруу чөйрөсүндө мындай процесстердин түзүмүн жана динамикасын визуалдык көрсөтүү үчүн ар кандай ыкмалар колдонуларын көрсөттү. Алсак, бир катар эмгектерде бул үчүн алгоритмдердин блок-схемалары да колдонулат, алар программалык системаларды иштеп чыгуунун тигил же бул тандалып алынган технология/методологиясына ылайык иштерди аткаруучу аткаруучулардын талап кылынган квалификациясы жөнүндө айрым комментарийлер менен гана толукталат. Заманбап, демек, татаал программалык камсыздоону иштеп чыгуу сөзсүз түрдө UML тилине негизделген объектке багытталган ыкманы колдонуу менен аны алдын ала долбоорлоодон башталышы керек. Ошондуктан, программалык камсыздоо долбоорун башкаруу процесстерин түзүү үчүн бул тилдин куралдарын колдонуу логикалык жактан негиздүү көрүнөт. Макалада программаларды иштеп чыгууну башкаруу процесстерин моделдөө үчүн UML куралдарын адаптациялоо эрежелери иштелип чыгат жана дисциплиналарды, фазаларды, Rational Unified Process технологиясынын контролдук чекиттерин көрсөтүү үчүн UML өткөөл диаграммаларын жана ишмердигин колдонууга мисал келтирет.

*Түйүндүү сөздөр:* системаларды визуалдык моделдөөнүн унификацияланган тили; программалык долбоорлорду башкаруу; программалык системаларды иштеп чыгуунун рационалдуу унификацияланган процессинин технологиясы; абал-фазалары; дисциплиналар-абалдар; milestone контролдук чекити.

## UML AS A SOFTWARE PROJECT MANAGEMENT TOOL

S. Ts. Manzhikova

**Abstract.** The article proposes to use the tools of the visual modeling language UML to plan the process of developing complex software in compliance with the conditions and requirements of the technology known as the Rational Unified Process (RUP). An analysis of relevant publications showed that in the field of organizing the management of software development processes, a variety of methods are used to visually represent the structure and dynamics of such processes. Thus, in a number of works, even block diagrams of algorithms are used for this purpose, which are only supplemented by some comments about the required qualifications of performers who work in accordance with a particular selected technology/methodology for developing software systems. The development of modern, and therefore complex software must necessarily begin with its preliminary design using an object-oriented method, which is based on the UML language. Therefore, it seems logically justified to use the tools of this language to build software project management processes. The article formulates the rules for adapting UML tools for modeling software development management processes and gives an example of using UML state transition and activity diagrams to represent disciplines, phases, and milestones of the Rational Unified Process technology.

**Keywords:** Unified Modeling Language; software project management; Rational Unified Process; phase-states; discipline-substates; checkpoint milestone.

**Введение.** Анализ публикаций показал, что в сфере организации управления процессами разработки программного обеспечения (ПО) используются самые разные способы визуального представления структуры и динамики таких процессов. Так, в работах [1–4] применяются изобразительные средства типа блок-схем алгоритмов, дополняемые некоторыми комментариями о квалификации исполнителей, выполняющих работы, предусмотренные той или другой выбранной технологией/методологией разработки программных систем (ПС).

Однако предварительное проектирование считается обязательным при разработке современных сложных ПС. Как правило, при этом используются методы объектно-ориентированного анализа и проектирования ПС – ООА и П ПС. Логическим следствием такого подхода является использование унифицированного объектно-ориентированного языка визуального моделирования/проектирования систем (и не только программных) UML.

В данной работе предлагается адаптация языка UML для моделирования/проектирования самих процессов разработки ПО. Аналогичный подход представлен в публикации [5], и он интересен тем, что применен для процесса создания/конфигурирования ПС в среде 1-С.

**Основные положения UML моделирования процессов управления разработкой программных проектов.**

1. Стадии, этапы или фазы процесса разработки ПО [1, 5–7] будем рассматривать как его состояния, достаточно продолжительные во времени. Такой взгляд позволяет рассматривать процесс разработки ПС как череду переходов из одного состояния в другое.

2. Состояния так же, как в UML, могут быть сложными, или композитными и простыми. Так, основные и вспомогательные рабочие процессы, составляющие дисциплины методологии RUP [6], могут рассматриваться как вложенные подсостояния фаз-состояний.

3. В свою очередь, каждая дисциплина-подсостояние [1, 5–7] реализуется по определенному алгоритму, который соответствует плану проекта ПО. Следовательно, действия/работы, выполняемые в рамках дисциплины-подсостояния, можно представить UML диаграммами деятельности/действий – Activity Diagrams.

4. Переходы между состояниями происходят, если свершилось какое-то событие или возникло определенное условие, или было произведено какое-то действие, т. е. спецификация перехода полностью соответствует семантике языка UML.

**Результаты UML моделирования унифицированного процесса RUP разработки программного обеспечения.** На рисунке 1 представлены фазы как состояния процесса разработки ПО в соответствии с технологией/методологией RUP. Каждая фаза-состояние начинается с входного действия *entry*, а заканчивается выполнением действия *exit* на выходе из этого состояния. Внутри состояния

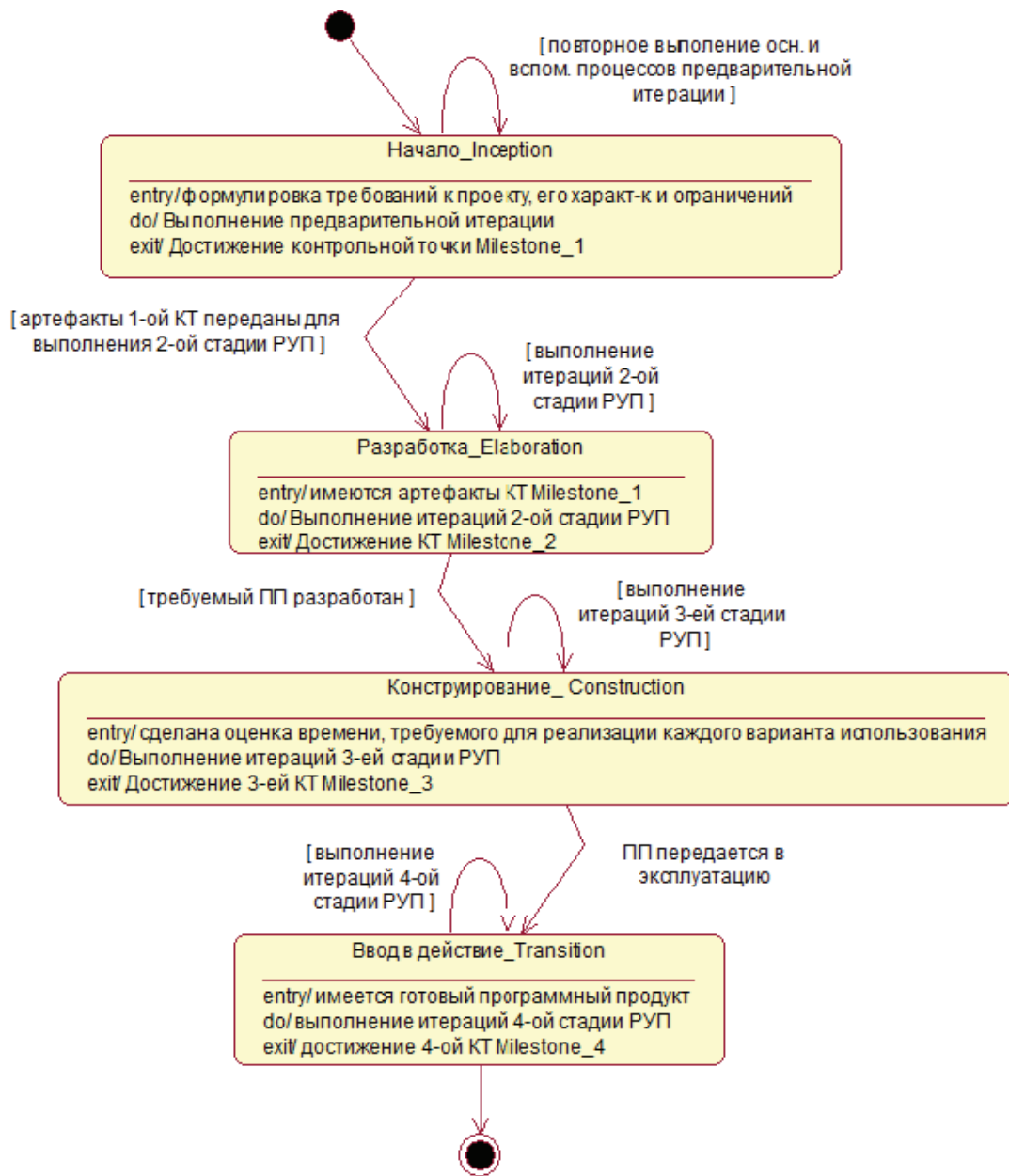


Рисунок 1 – Фазы-состояния рационального унифицированного процесса (РУП или RUP)

выполняется действие *do*. Согласно концепции RUP, каждая его фаза может выполняться итеративно, что показано стрелкой-переходом на само это состояние; такой переход совершается при условии, показанном в квадратных скобках рядом с переходом. Переход же к следующей фазе разработки ПС происходит только в случае, если достигнута контрольная точка (КТ) *Milestone\_1*, т. е. выполнены все условия завершенности фазы разработки. Эти условия показываются на диаграммах, представленных ниже.

Каждая фаза RUP является композитным состоянием, включающим вложенные подсостояния. В качестве таких подсостояний рассматриваются основные и вспомогательные процессы [6], которые принято называть дисциплинами RUP (рисунок 2). Дисциплины «Требования», «Анализ и проектирование», «Реализация», «Тестирование» имеют место в каждой фазе, поэтому в статье приводится только одна диаграмма вложенных состояний для фазы Начало *Inception*.

Переходы с предыдущего подсостояния в последующее специфицированы событиями, например, дисциплина «Требования» выполнена или Контр-точка\_1 достигнута.

Итеративные повторения выполняемых дисциплин специфицированы условиями, заключенными в квадратные скобки, и действиями, запланированными в рамках итераций и записанными с условием через слеш /, например, [*корректировка арх-ры и кода ПС*]/итерации\_3,4.

Семантика и графические возможности UML диаграмм деятельности – Activity Diagrams – позволяют показать следующие аспекты процесса разработки ПО (рисунки 3–6):

- что надо иметь/сделать, чтобы начать выполнение конкретной фазы: для этого выделена «дорожка» *Входная информация*;
- какие конкретно работы/действия необходимо выполнить: «дорожка» *Деятельность*;
- кто должен выполнять спланированные работы/действия: «дорожка» *Исполнитель*;
- что будет результатом этих работ/действий отражается в «дорожке» *Вых. информация / Артефакты*;
- какая цель диктует необходимость получения/оформления *Выходной информации/Артефактов*: «дорожка» *Правила/регламент*.

Компоненты *Вх. Информации* представлены объектами, необходимыми для выполнения *Деятельности*. Число и типы этих объектов изменяются от фазы к фазе.

Наименования *Деятельностей* на рисунках 3–6 остаются теми же самыми, но качественно меняются их направленность и содержание, и это показано на диаграммах рисунков 3–6.

С каждой последующей фазой увеличивается число *Исполнителей* проекта, т. е. UML диаграммы процесса разработки ПО помогают уточнить состав команды разработчиков для каждой стадии/фазы.

Конкретный вид *Вых. информации / Артефактов* закреплен за конкретным *Исполнителем*, что облегчает четкое распределение обязанностей между членами команды разработчиков программного проекта.

В работе [6] строго формулируются условия, при которых фаза процесса RUP разработки ПС может считаться завершенной, а контрольная точка КТ *Milestone* – достигнутой. Например, КТ *Milestone\_3* может считаться достигнутой, если выполняются следующие условия перехода к ней:

- “[*Расхождения реальных доходов с предполагаемыми приемлемо*]”;
- “[*Программный продукт достаточно стабилен и качественен для распространения среди пользователей*]”;
- “[*Заинтересованные стороны одобрили и готовы к внедрению программного продукта*]”.

В фазе *Ввод в действие Transition Деятельности Определение требований и Анализ* не проводятся, что наглядно показано на UML диаграмме (рисунок 6).

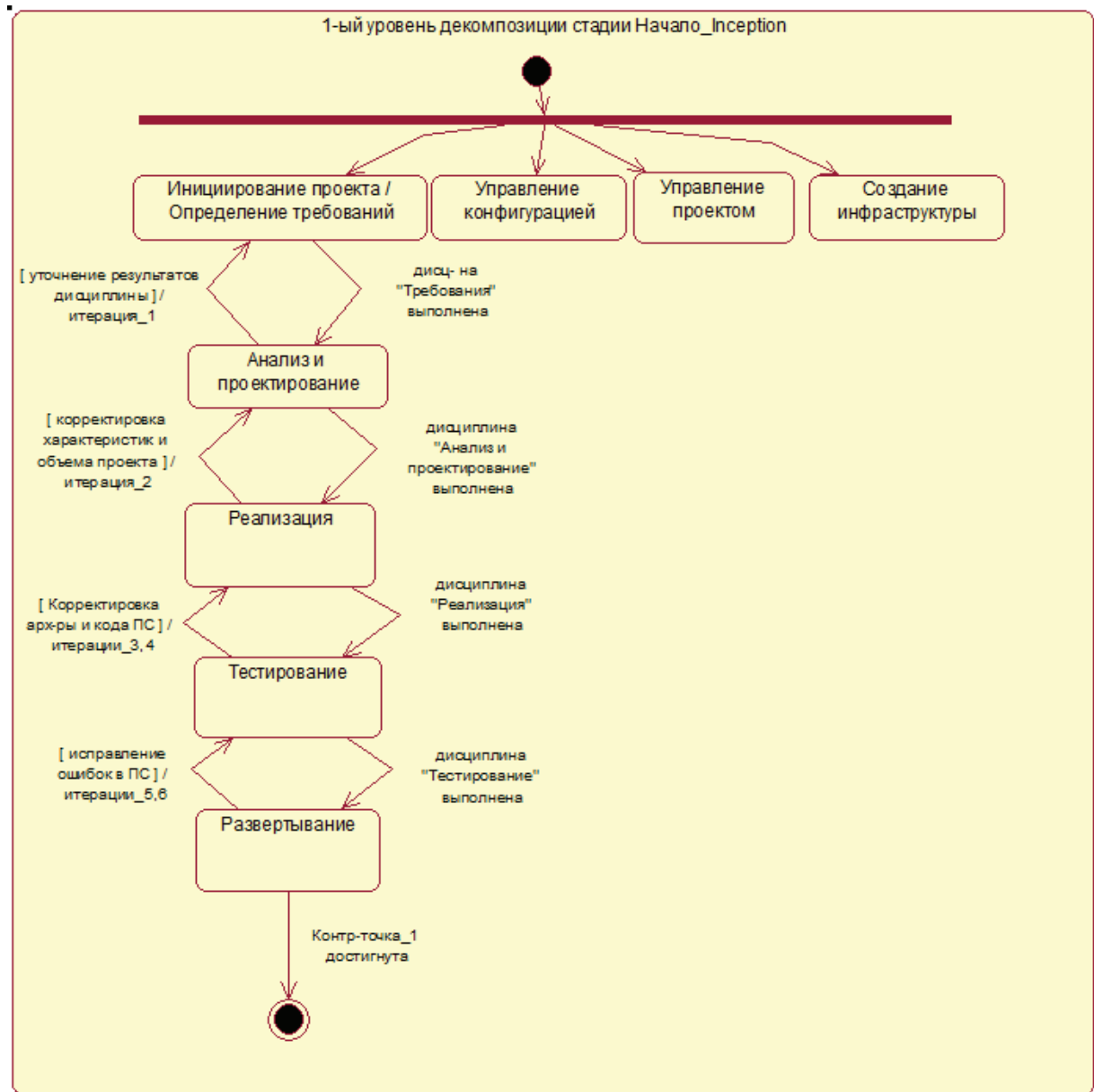


Рисунок 2 – Основные и вспомогательные процессы, составляющие дисциплины-подсостояния

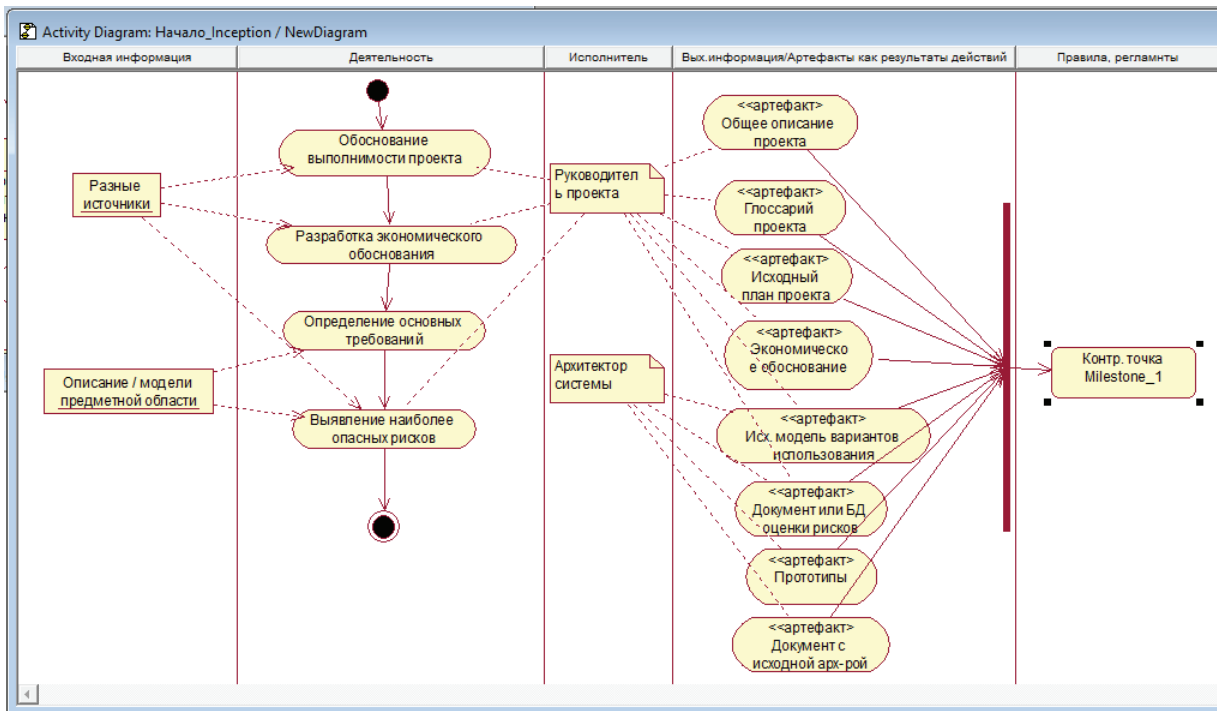


Рисунок 3 – Работы/действия, выполняемые в фазе *Начало\_Inception*

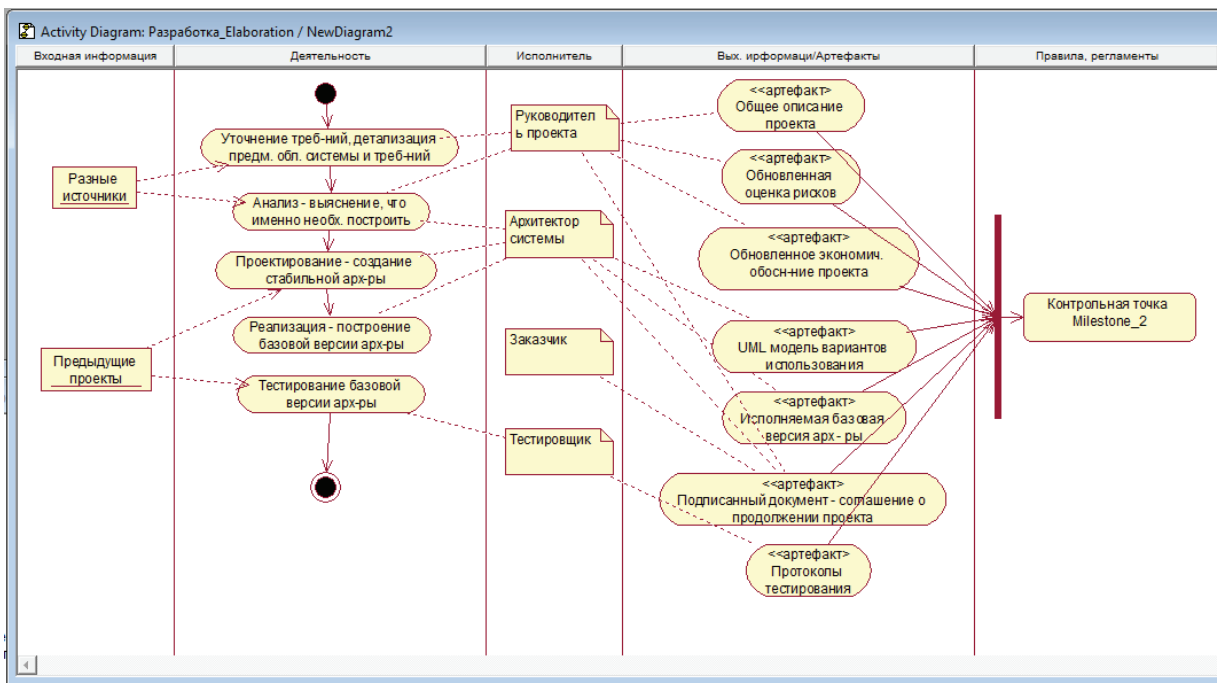


Рисунок 4 – Работы/действия, выполняемые в фазе *Разработка\_Elaboration*

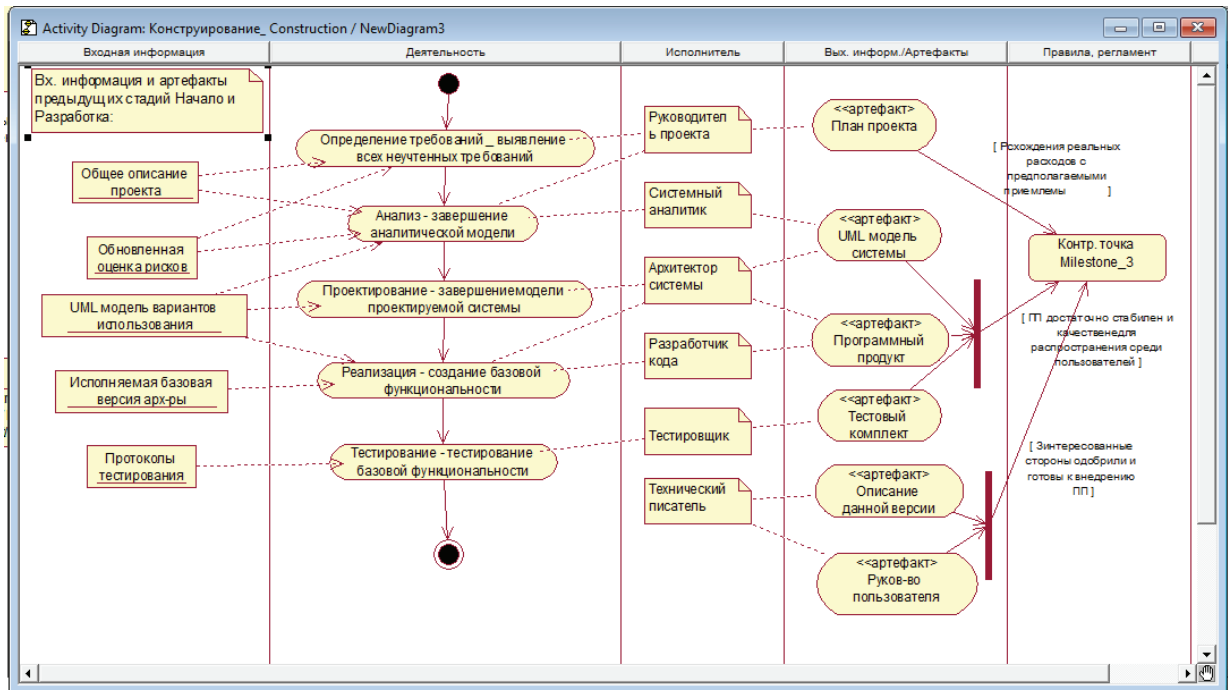


Рисунок 5 – Работы/действия, выполняемые в фазе *Конструирование\_Construction*

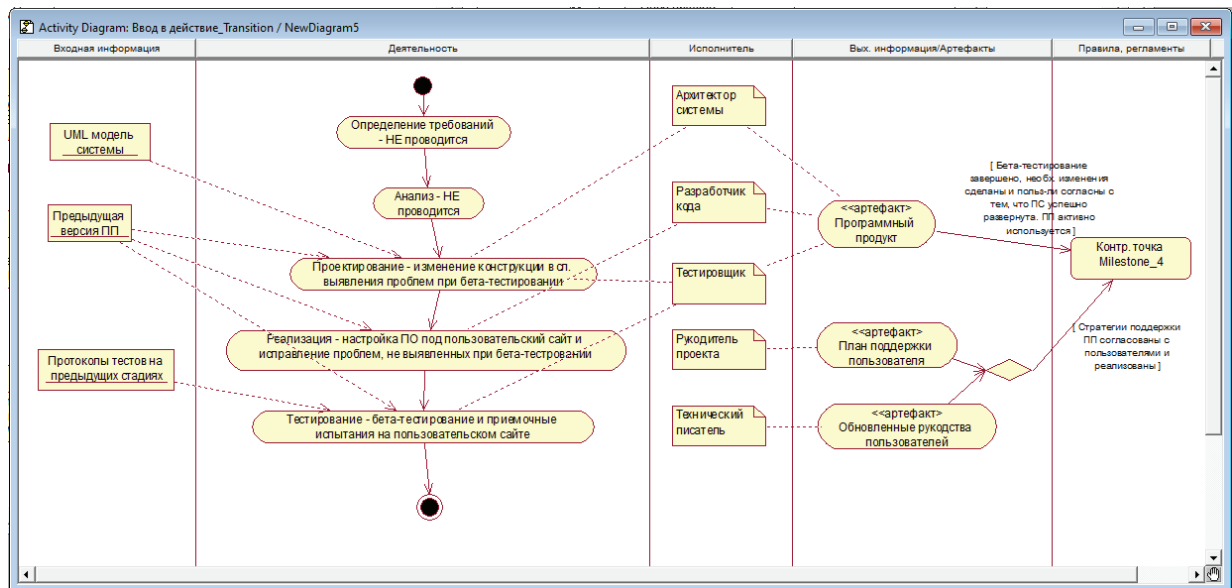


Рисунок 6 – Работы/действия, выполняемые в фазе *Ввод в действие\_Transition*

**Заключение.** Сформулированы основные положения, позволяющие применять UML диаграммы переходов состояний – State Chart Diagram – и диаграммы деятельности – Activity Diagram – для моделирования процессов разработки ПО.

В работе приведен пример UML моделирования фаз, дисциплин и рабочих процессов унифицированного процесса RUP фирмы Rational.

Показано, что такая визуализация способствует наглядному представлению многомерного процесса разработки программного проекта, потому что семантика применяемых диаграмм охватывает большинство аспектов таких процессов:

- что надо иметь/сделать, чтобы начать выполнение конкретной фазы;
- какие конкретно работы/действия необходимо выполнить;
- какой специалист должен выполнять спланированные работы/действия;
- что должно быть результатом этих работ/действий;
- какая цель диктует необходимость получения/оформления выходной информации/артефактов.

Результаты работы могут использоваться в практике программных проектов, а также в преподавании соответствующих дисциплин по направлениям «Программная инженерия» и «Управление программными проектами».

Поступила: 16.11.2023; рецензирована: 30.11.2023; принята: 04.12.2023.

#### *Литература*

1. Кулямин В. Унифицированный процесс разработки и экстремальное программирование. URL: [https://intuit.ru/studies/courses/64/64/print\\_lecture/1870](https://intuit.ru/studies/courses/64/64/print_lecture/1870) (дата обращения: 15.09.2023).
2. Евстратов М. Опыт перехода с Waterfall на методологию RUP для реализации больших ИТ проектов. URL: <https://habr.com/ru/post/329572/> (дата обращения: 15.09.2023).
3. Холодков А. Разработка ПО: пример бизнес-процесса из практики. URL: <https://xn----8sbwjfice2afgci1k.xn--p1ai/it/?p=130> (дата обращения: 5.09.2023).
4. Управление проектом разработки web-приложения с использованием методологии RUP. URL: <https://ru.essays.club/Точные-науки/Программирование-и-компьютеры/Управление-проектом-разработки-79906.html> (дата обращения: 15.09.2023).
5. Наумов С. Практика применения UML для проектирования бизнес-процессов и информационных систем. URL: <https://infostart.ru/1c/articles/662084/> (дата обращения: 10.09.2023).
6. Арлоу Д. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Д. Арлоу, И. Нейштадт; пер. с англ. 2-е изд. СПб.: Символ Плюс, 2007. 624 с.
7. Орлов С.А. Программная инженерия: учебник для вузов / С.А. Орлов. 5-е изд., обновл. и доп. СПб., 2018. 640 с.