

UML-технологии для проектирования программного обеспечения

1. Унифицированный процесс разработки программного обеспечения

Владимир Юрьевич Романов,
Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
vromanov@cs.msu.su,
romanov.rvy@yandex.ru

Предыстория унифицированного процесса

Rational Unified Process (RUP)

- **RUP** был создан фирмой **Rational** – разработчиком популярного CASE-инструмента **Rational Rose** и ряда других инструментов для разработки ПО
- В 2003 году фирма **Rational** вошла в фирму **IBM** как структурное подразделение
- Rational Unified Process был предложен для стандартизации в консорциум фирм Object Management Group (OMG).

Цель стандартизации – возможность поддержки *унифицированного процесса разработки ПО* различными инструментами на основе унифицированной метамодел*и процесса* разработанной консорциумом OMG.

Стандартизация унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Каталог стандартов консорциума Object Management Group связанных со стандартом на язык UML:

http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML

2. *Мета модель* процесса разработки программного обеспечения
Software **P**rocess **E**ngineering **M**etamodel (**SPEM**)

<http://www.omg.org/spec/SPEM/2.0/>

Стандарт SPEM 2.0 консорциума OMG

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0

Release Date: April 2008

Normative

OMG document number	Explanation	Format	URL
formal/2008-04-01	Formal specification	PDF	http://www.omg.org/spec/SPEM/2.0/PDF
formal/2008-04-01	Formal specification	PS	http://www.omg.org/spec/SPEM/2.0/PS
ptc/07-08-08 ptc/07-08-09	collection of schema files	CMOF XMI	http://www.omg.org/spec/SPEM/20070801/SPEM2-Process-Behavior-Content.merged.cmoF
			http://www.omg.org/spec/SPEM/20070801/Infrastructure.cmoF
			http://www.omg.org/spec/SPEM/20070801/LM.cmoF
			http://www.omg.org/spec/SPEM/20070801/SoftwareProcessEngineeringMetamodel.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM2-Method-Content.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM2-Process-Behavior-Content.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM2.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM2.merged.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM2-Method-Content.merged.cmoF
			http://www.omg.org/spec/SPEM/20070801/SPEM 2.0 Base Plugin Profile.xml
			http://www.omg.org/spec/SPEM/20070801/SPEM 2.0 UML 2 Profile.xml

Состав стандарта SPEM 2.0

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

OMG document number	Explanation	Format	URL
formal/2008-04-01	Formal specification	PDF	http://www.omg.org/spec/SPEM/2.0/PDF
formal/2008-04-01	Formal specification	PS	http://www.omg.org/spec/SPEM/2.0/PS
ptc/07-08-08 ptc/07-08-09	collection of schema files	CMOF XMI	http://www.omg.org/spec/SPEM/20070801/SPEM2-Process-Behavior-Content.merged.cmf http://www.omg.org/spec/SPEM/20070801/Infrastructure.cmf http://www.omg.org/spec/SPEM/20070801/LM.cmf http://www.omg.org/spec/SPEM/20070801/SoftwareProcessEngineeringMetamodel.cmf http://www.omg.org/spec/SPEM/20070801/SPEM2-Method-Content.cmf http://www.omg.org/spec/SPEM/20070801/SPEM2-Process-Behavior-Content.cmf http://www.omg.org/spec/SPEM/20070801/SPEM2.cmf http://www.omg.org/spec/SPEM/20070801/SPEM2.merged.cmf http://www.omg.org/spec/SPEM/20070801/SPEM2-Method-Content.merged.cmf http://www.omg.org/spec/SPEM/20070801/SPEM 2.0 Base Plugin Profile.xmi http://www.omg.org/spec/SPEM/20070801/SPEM 2.0 UML 2 Profile.xmi

Реализация унифицированного процесса как подпроекта Eclipse

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Eclipse Process Framework Project (EPF)

<http://www.eclipse.org/epf/>

Цель проекта:

способствовать созданию настраиваемого
фреймворка разработки программного обеспечения (SDE)

с примерным содержимым и инструментами процесса

поддерживающим широкий спектр проектов и стилей
разработки

Пример – Open UP

OpenUP - Mozilla Firefox: IBM Edition

file:///C:/temp/OpenUP/Publish/index.htm

OpenUP

Where am I | Tree Sets | OpenUP

- Introduction to OpenUP
- Getting Started
- OpenUP Disciplines
- OpenUP Work Products
- OpenUP Roles
- OpenUP lifecycle
- About
- OpenUP Copyright

What is OpenUP?

OpenUP is a lean Unified Process that applies iterative and incremental approaches within a structured lifecycle. OpenUP embraces a pragmatic, agile philosophy that focuses on the collaborative nature of software development. It is a tools-agnostic, low-ceremony process that can be extended to address a broad variety of project types.

The diagram illustrates the OpenUP process across three layers:

- Micro-Increment (Personal Focus):** Shows a 'Work Item' (Days) leading to an 'Increment' (gear).
- Iteration Lifecycle (Team Focus):** Shows an 'Iteration Plan' (Weeks) leading to an 'Iteration' (gear) which produces a 'Demo-able or Shippable Build' (cube).
- Project Lifecycle (Stakeholder Focus):** Shows a 'Project Plan' (Months) leading to a timeline with phases: Inception, Elaboration, Construction, and Transition. A graph shows 'Risk' (red line) decreasing and 'Value' (green line) increasing over time.

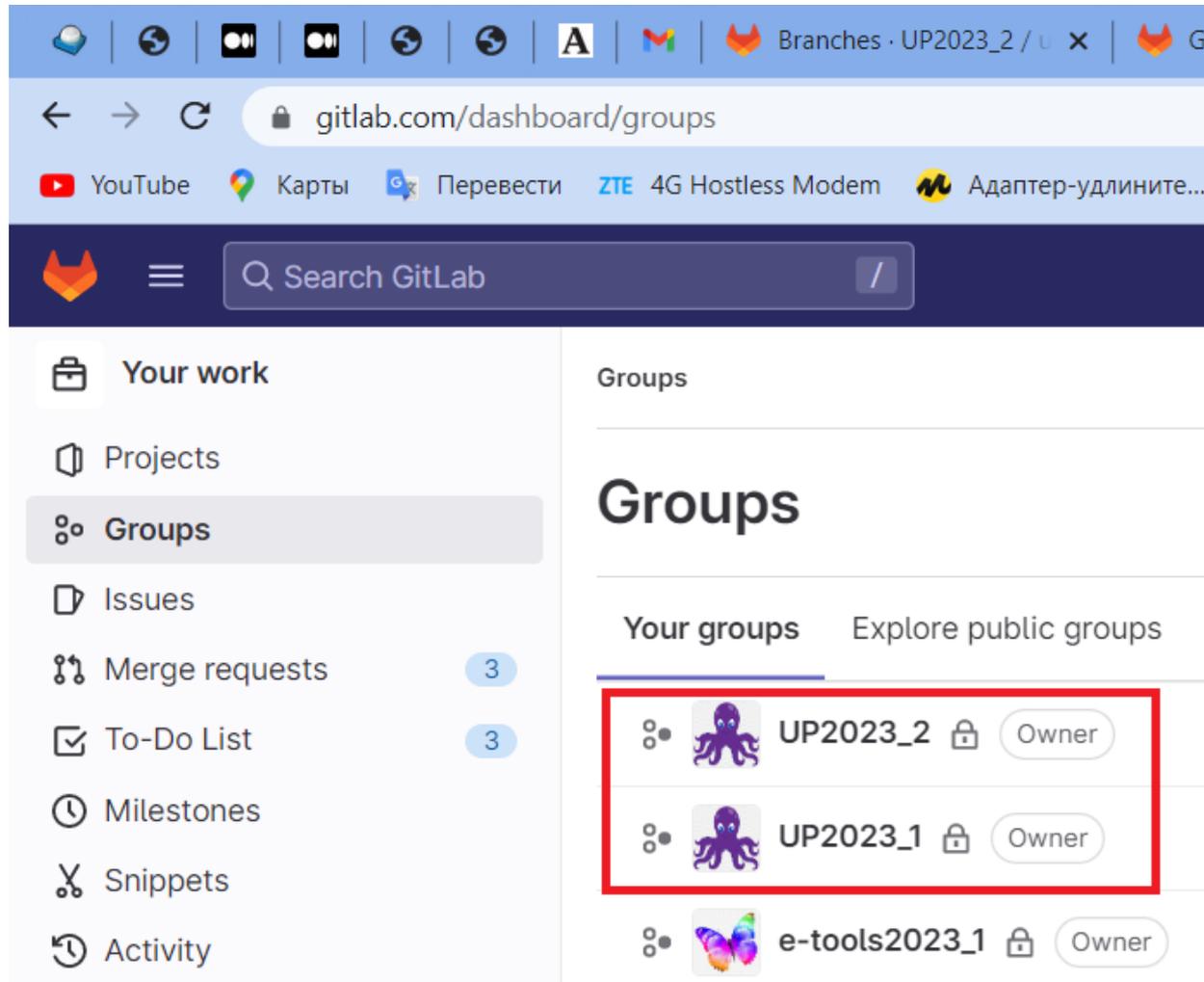
OpenUP layers: micro-increments, iteration lifecycle and project lifecycle

Personal effort on an OpenUP project is organized in **micro-increments**. These represent short units of work that produce a steady, measurable pace of project progress (typically measured in hours or a few days). The process applies intensive collaboration as the system is incrementally developed by a committed, self-organized team. These micro-increments provide an extremely short feedback loop that drives adaptive decisions.

Группы практикума с использованием CASE-инструмента *Papyrus*

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



The screenshot shows the GitLab dashboard interface. The browser address bar displays `gitlab.com/dashboard/groups`. The left sidebar contains navigation options: 'Your work', 'Projects', 'Groups' (highlighted), 'Issues', 'Merge requests' (3), 'To-Do List' (3), 'Milestones', 'Snippets', and 'Activity'. The main content area is titled 'Groups' and features a sub-header 'Your groups' and a link 'Explore public groups'. A list of groups is displayed, with the first two groups, 'UP2023_2' and 'UP2023_1', enclosed in a red rectangular box. Each group entry includes a group icon, the group name, a lock icon, and the role 'Owner'.

Group Name	Role
UP2023_2	Owner
UP2023_1	Owner
e-tools2023_1	Owner

Назначение унифицированного процесса

1. Предоставляет руководство по упорядочению деятельности команд разработчиков ПО
2. Управляет задачами решаемыми конкретными разработчиками ПО и командой в целом
3. Описывает какие артефакты (UML-модели, программы, документы) должны быть разработаны
4. Предоставляет критерии для оценки и измерения продуктов проекта и деятельности проекта
5. Вся разработка ведется **на основе UML-моделей** разрабатываемого программного обеспечения

Назначение унифицированного процесса

1. Предоставляет руководство по упорядочению деятельности команд разработчиков ПО
2. Управляет задачами решаемыми конкретными разработчиками ПО и командой в целом
3. Описывает какие артефакты (UML-модели, программы, документы) должны быть разработаны
4. Предоставляет критерии для оценки и измерения продуктов проекта и деятельности проекта
5. Вся разработка ведется **на основе UML-моделей** разрабатываемого программного обеспечения

Применимость Унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Унифицированный процесс – это **типовой процесс** для:

- Для широкого класса проектируемых программных систем
- Для различных областей приложений
- Для различных типов организаций
- Для различных уровней компетентности
- Для различных размеров проектов

Отличительные особенности Унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Управляется прецедентами (use case driven)
2. Ориентирован на раннюю разработку архитектуры
3. Итеративный и инкрементальный

Управление прецедентами в Унифицированном процессе (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Для построения программной системы необходимо знание потребностей *пользователя*
- *Пользователь (user)* – человек или другая программная система расположенные вне разрабатываемой системы
- *Прецедент (use case)* – взаимодействие пользователя и системы
- Все прецеденты вместе – это *модель прецедентов (Use Case Model)*
- Отличие UCM от функциональной спецификации – привязка функциональности к актерам

Управление прецедентами в Унифицированном процессе (2)

- Модель прецедентов (UCM) порождает серию других **UML-моделей** соответствующих UCM:
 - **Модель анализа** (*Analysis Model*)
 - **Модель проектирования** (*Design Model*)
 - **Модель реализации** (*Implementation Model*)
 - **Модель тестирования** (*Testing Model*)

Управление прецедентами в Унифицированном процессе (3)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Процесс разработки следует *потоку (flow)*
- Процесс разработки проходит через *потоки работ (workflows)*
- Прецеденты **влиять на создание** архитектуры программной системы
- Архитектура программной системы затем **влияет на выбор** прецедентов использования при проектировании и реализации этой системы

Унифицированный процесс ориентирован на разработку архитектуры системы (1)

- Существует *аналогия архитектуры* в строительстве и разработке программного обеспечения
- **Архитектура в строительстве** (точки зрения на здание)
 - Структура
 - Электрификация
 - Водоснабжение
 - Обогрев
 - Сервис
- *Архитектура в строительстве позволяет видеть все здание **до** начала строительства*

Унифицированный процесс ориентирован на разработку архитектуры системы (2)

Архитектура в разработке программного обеспечения

- *Порождается из нужд пользователя – модели прецедентов (Use Case Model)*
- *Зависит также от:*
 - **Платформы:**
 - *архитектуры компьютера;*
 - *операционной системы;*
 - *СУБД;*
 - *связанности компьютеров в сети*
 - **Переиспользуемых компонент:**
 - *framework для построения интерфейса пользователя*
 - *соглашению по внедрению системы (deployment)*
 - *нефункциональным требованиям (производительность, память, ...)*

Унифицированный процесс ориентирован на разработку архитектуры системы (3)

Архитектура в разработке программного обеспечения

- Это *точка зрения (view)* на программную систему в целом, в которой:
 - Существенные свойства системы сделаны *видимыми*
 - Несущественные свойства системы *скрыты*
 - *Архитектор* обладает достаточным опытом, что бы определить что существенно, а что не существенно

Унифицированный процесс ориентирован на разработку архитектуры системы (5)

Архитектура в разработке программного обеспечения

- *Архитектор создает черновик (draft) архитектуры связанный не с прецедентами, а с платформой*
- *Архитектор далее работает с набором ключевых прецедентов.*
- *Каждый ключевой прецедент описывается и реализуется с использованием понятий: подсистем, классов, компонент*
- *Разработанная архитектура затем используется для выбора и реализации оставшихся прецедентов*

Унифицированный процесс ориентирован на разработку архитектуры системы (4)

Архитектура в разработке программного обеспечения

- Унифицированный процесс позволяет **архитектору** сконцентрироваться на правильных целях:
 - Понимаемости системы
 - Устойчивости системы к последующим изменениям
 - Переиспользуемости компонент системы
- Архитектор находит 5-10% ключевых прецедентов

Унифицированный процесс итеративный и инкрементальный (1)

- *Разработка ПО может продолжаться от нескольких месяцев до нескольких лет*
- *По этой причине разработка ПО разделяется на несколько **мини-проектов***
- *Каждый **мини-проект** называется **итерацией***
- *Итерация выбирается и выполняется по плану*
- *Итерация проходит через **потоки работ (workflows)***
- *Результат итерации - **инкремент***

Унифицированный процесс итеративный и инкрементальный (2)

- *Необходимость итерации*
 - *Итерация имеет дело с группой прецедентов расширяющих использование системы*
 - *Итерация имеет дело с наиболее важными **рисками***
- *Итерация проходит через более мелкие шаги - **потоки работ:***
 - *Поток работ анализа*
 - *Поток работ проектирования*
 - *Поток работ реализации*
 - *Поток работ тестирования*

Унифицированный процесс итеративный и инкрементальный (3)

Достоинства итеративного процесса

- *Уменьшается стоимость риска для одного инкремента. В случае неудачи теряется только одна итерация*
- *Уменьшается риск не поставки системы в заданные сроки. Риски определяются в начале проекта*
- *Ускоряется разработка ПО. Разработчик имеет ясную цель на короткий период времени*
- *По мере реализации итераций уточняются требования к системе*

Унифицированный процесс итеративный и инкрементальный (3)

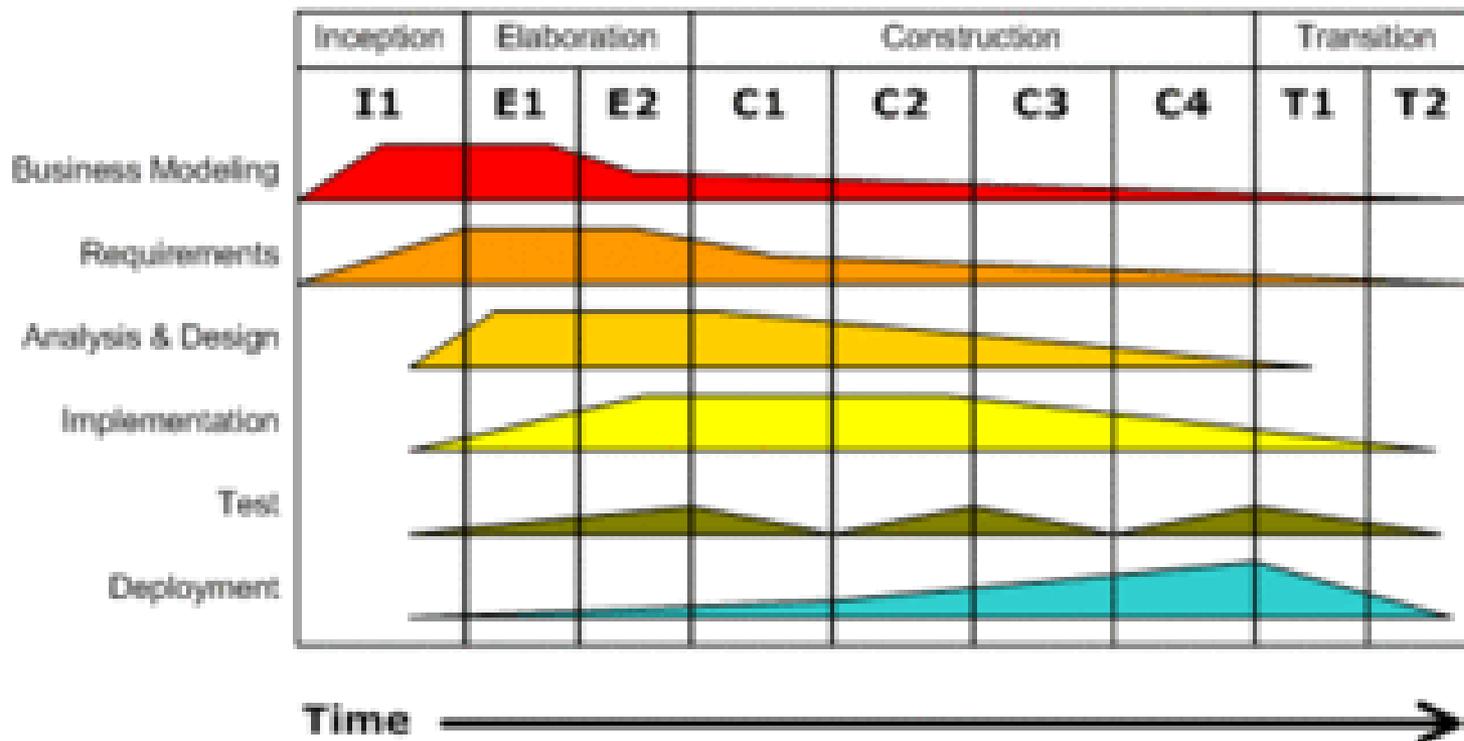
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Фазы итеративного процесса

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Процесс разработки, ориентированный на
управление сценариями использования
системы.

Процесс управляемый прецедентами

Что определяют прецеденты

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Поиск и описание классов*
- *Поиск и описание подсистем и интерфейсов*
- *Поиск и описание прецедентов тестирования*
- *Планирование итераций разработки*
- *Системную интеграцию*

Процесс управляемый прецедентами. Цели сбора требований к системе

1. Определение существенных требований.

Существенные те, которые принесут пользователю осязаемый и значимый результат.

2. Представление в форме удобной для:

- пользователей*
- заказчиков*
- разработчиков*

с использованием понятного для всех языка

Процесс управляемый прецедентами. Построение модели прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Выявление *категорий пользователей*.
2. Представление категорий с помощью *актеров* языка UML
3. Для каждого актера определяются *прецеденты* – последовательности действий выполняемых системой для получения актером необходимого результата
4. *Модель прецедентов* – полный набор актеров и прецедентов

Процесс управляемый прецедентами. Построение модели анализа (1)

1. *Модель анализа* - описание структуры системы с использованием *концептуальных классификаторов*
2. Отдельная модель сопровождая для больших систем
3. Первая модель в понятиях привычных разработчику системы (классификаторы, пакеты, ...)
4. Используется также и для сбора требований

Процесс управляемый прецедентами. Построение модели анализа (2)

5. *Нет привязки к особенностям реализации*

- *к языку программирования*
- *к операционной системе*
- *системе управления базами данных*
- *библиотекам используемым для построения интерфейса пользователя*

6. *Составляет 20% от объема модели проектирования (Design Model)*

Процесс управляемый прецедентами. Построение модели проектирования (1)

1. Для *модели проектирования (design model)* строится иерархия классов и интерфейсов
2. Описываются детально отношения ассоциации
3. Описываются отношения зависимости
4. Детально с использованием классификаторов и с помощью диаграмм поведения языка UML описываются реализации прецедентов
5. Затем *модель проектирования* отображается в *модель реализации (Implementation Model)*

Процесс управляемый прецедентами. Построение модели проектирования (2)

4. *Выполняется привязка к конкретным инструментам разработки:*
 - *языку программирования*
 - *операционной системе*
 - *системе управления базами данных*
 - *библиотекам для построения интерфейса пользователя*

5. *Классы группируются в подсистемы, между которыми определяются интерфейсы*

6. *Строится **модель внедрения (deployment model)** описывающая распределение системы по вычислительным узлам*

Процесс управляемый прецедентами. Построение модели реализации

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. *Разбиение системы на компоненты*
 - *файлы исходных текстов*
 - *jar-файлы*
 - *war-файлы*
 - *exe-файлы*
 - *DLL-файлы*

2. *Распределение компонент по вычислительным узлам и директориям файловой системы*

3. *Определение отношений зависимости между компонентами*

Процесс управляемый прецедентами. Построение модели тестирования (1)

1. *Модель тестирования (Testing Model)*
 - для проверки реализации функциональности определенной в модели прецедентов
 - для проверки нефункциональных требований к системе
2. Модель тестирования содержит *прецеденты тестирования (Test Cases)*. Из прецедентов получаются:
 - входные данные
 - условия выполнения прецедентов
 - результаты

Процесс управляемый прецедентами. Построение модели тестирования (2)

4. *Прецеденты тестирования (Test Cases) получают сразу после получения прецедентов использования (Use Cases).*

Следовательно *прецеденты использования* возможно планирование *тестирования «черного ящика»*

5. *После получения модели проектирования возможно тестирование «белого ящика».*

Каждая ветвь в реализации прецедента использования определяет отдельный прецедент тестирования белого ящика.

Отличительные особенности Унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

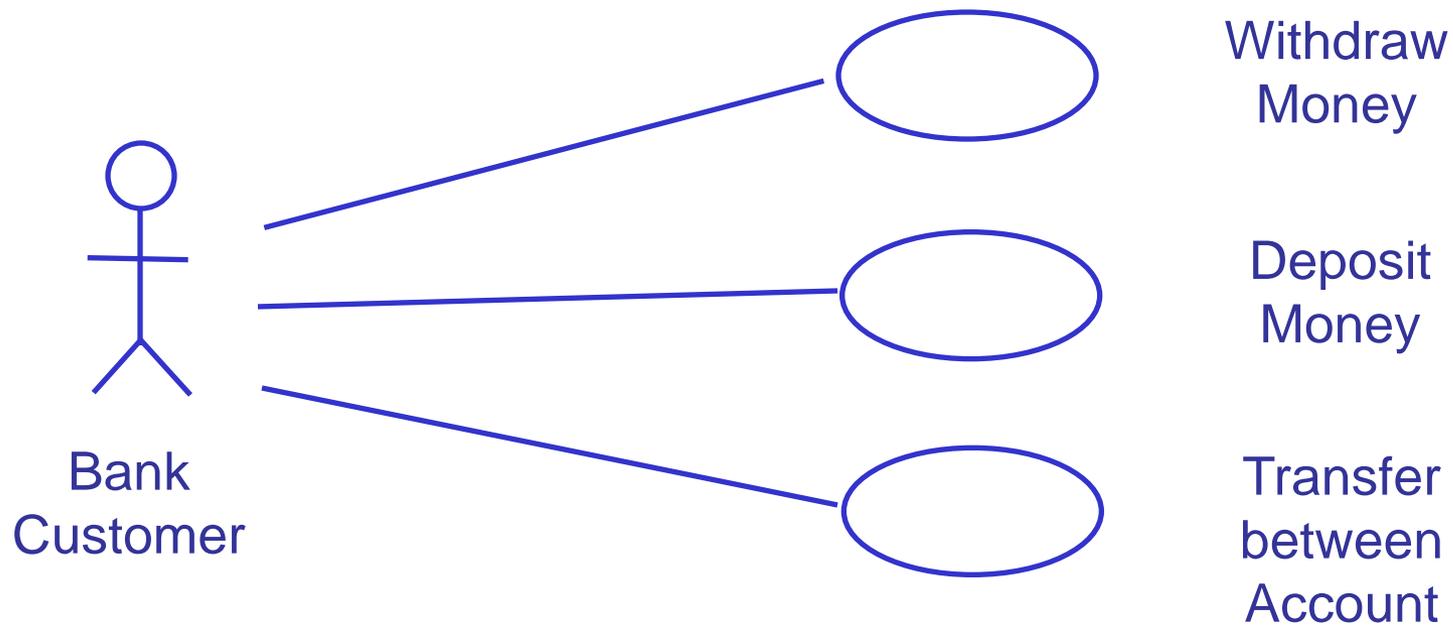
-  1. Управляется прецедентами (use case driven)
2. Ориентирован на раннюю разработку архитектуры
3. Итеративный и инкрементальный

Построение модели прецедентов. Пример

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Функциональные требования:



Нефункциональные требования к *WithdrawMoney*:

время отклика для *BankCustomer* не более 30 сек в 95% случаев

Построение модели анализа. Пример

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Для всех прецедентов текущей итерации строится система классификаторов и отношений между ними

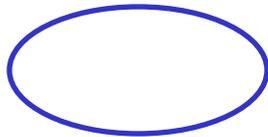
- *Какие классификаторы необходимы для реализации прецедентов*
- *Какие отношения ассоциации между классификаторами необходимы для реализации прецедента*

Построение модели анализа. Выбор классификаторов.

МГУ им. М.В.Ломоносова. Факультет ВМК.

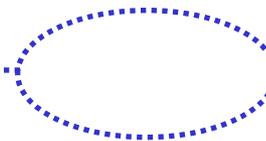
Романов Владимир Юрьевич ©2025

Модель
прецедентов



Withdraw
Money

Модель
анализа



<<trace>>



Withdraw
Money



Dispenser



Cashier
Interface



Withdraw



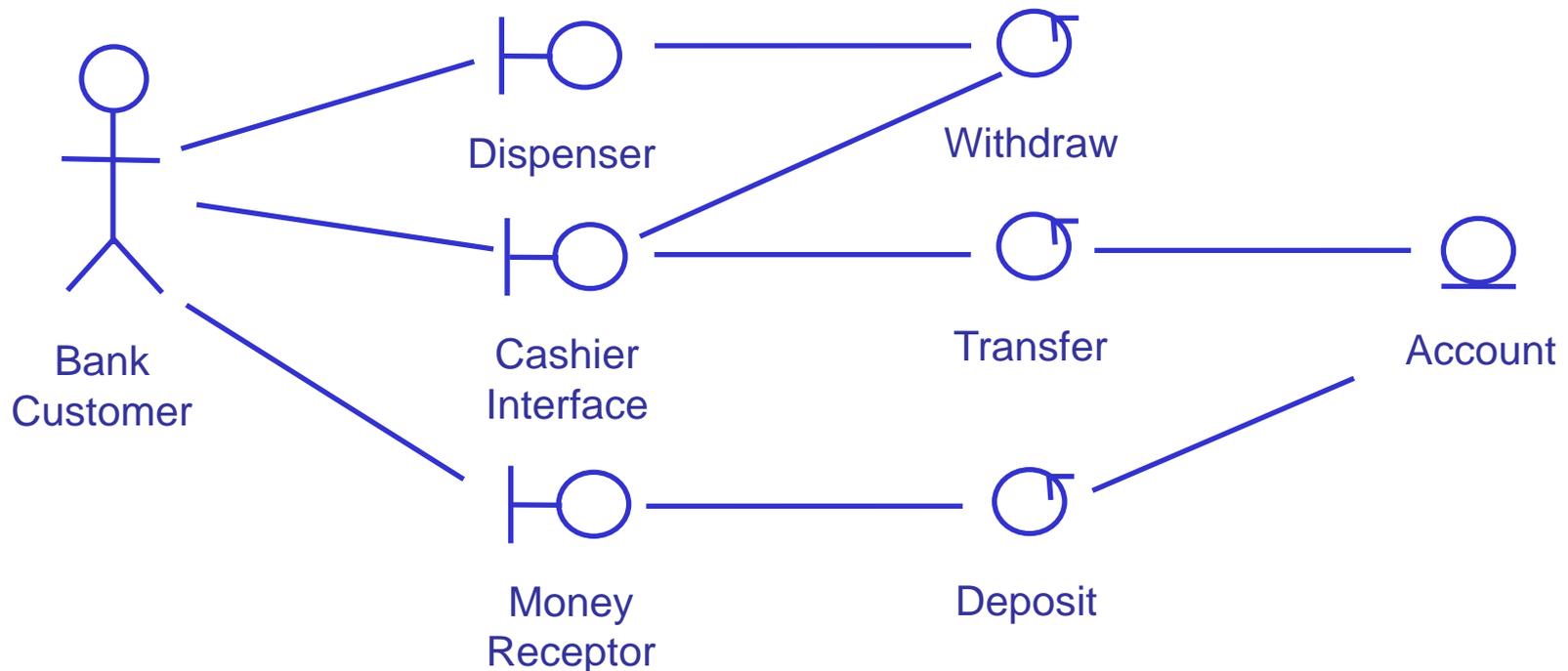
Account

Построение модели анализа. Определение отношений ассоциации

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Модель анализа

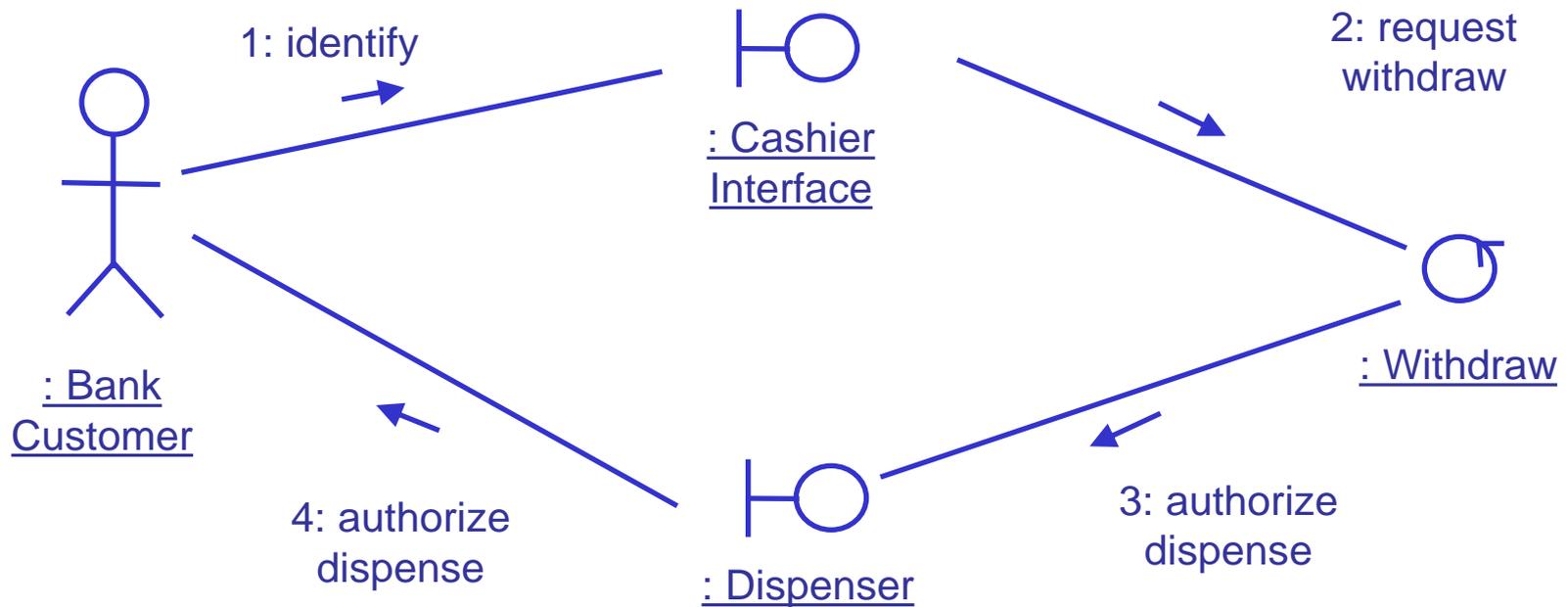


Построение модели анализа. Описание реализации прецедента

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Модель анализа

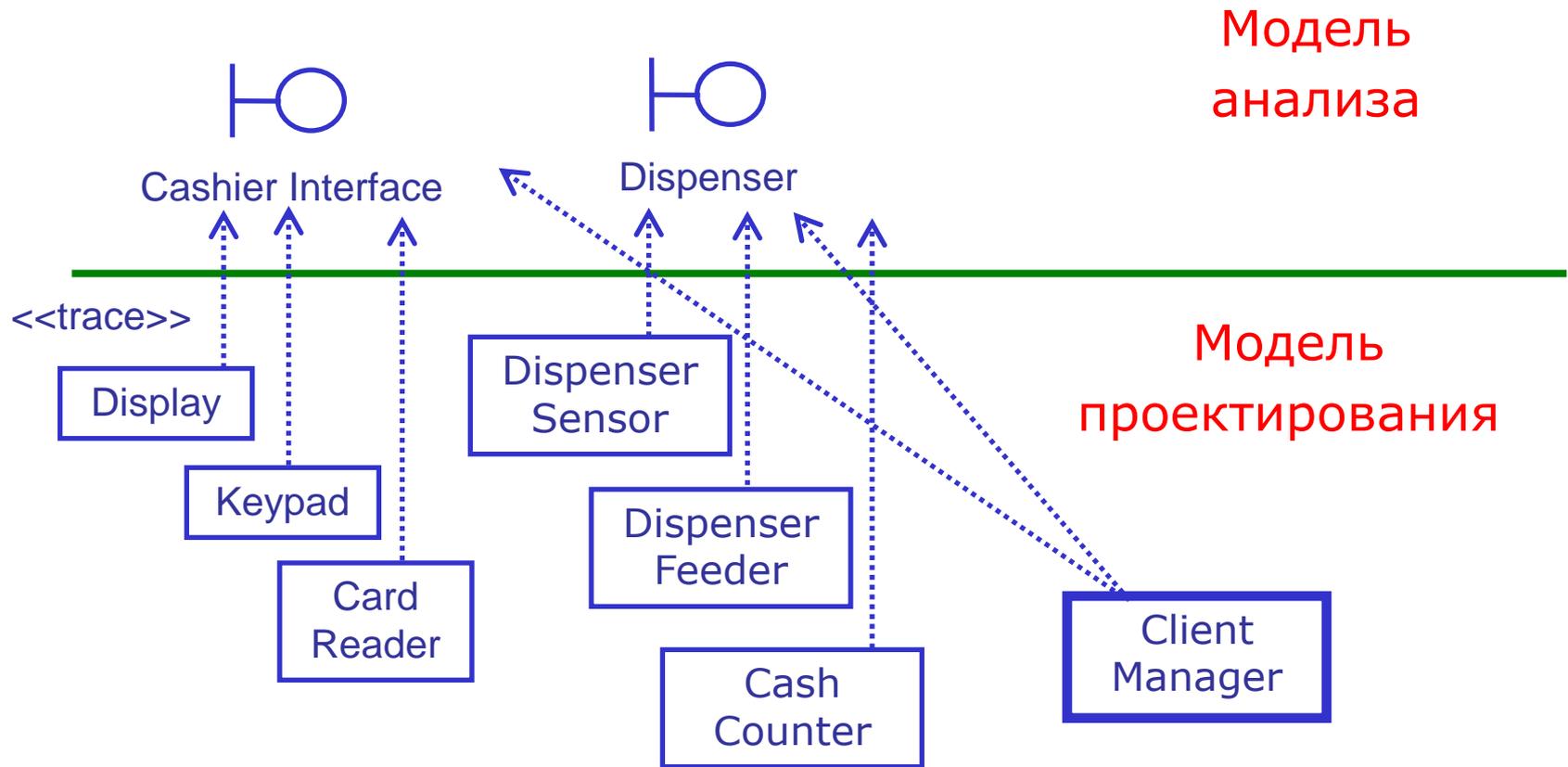


Построение модели проектирования.

Определение классов модели проектирования (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

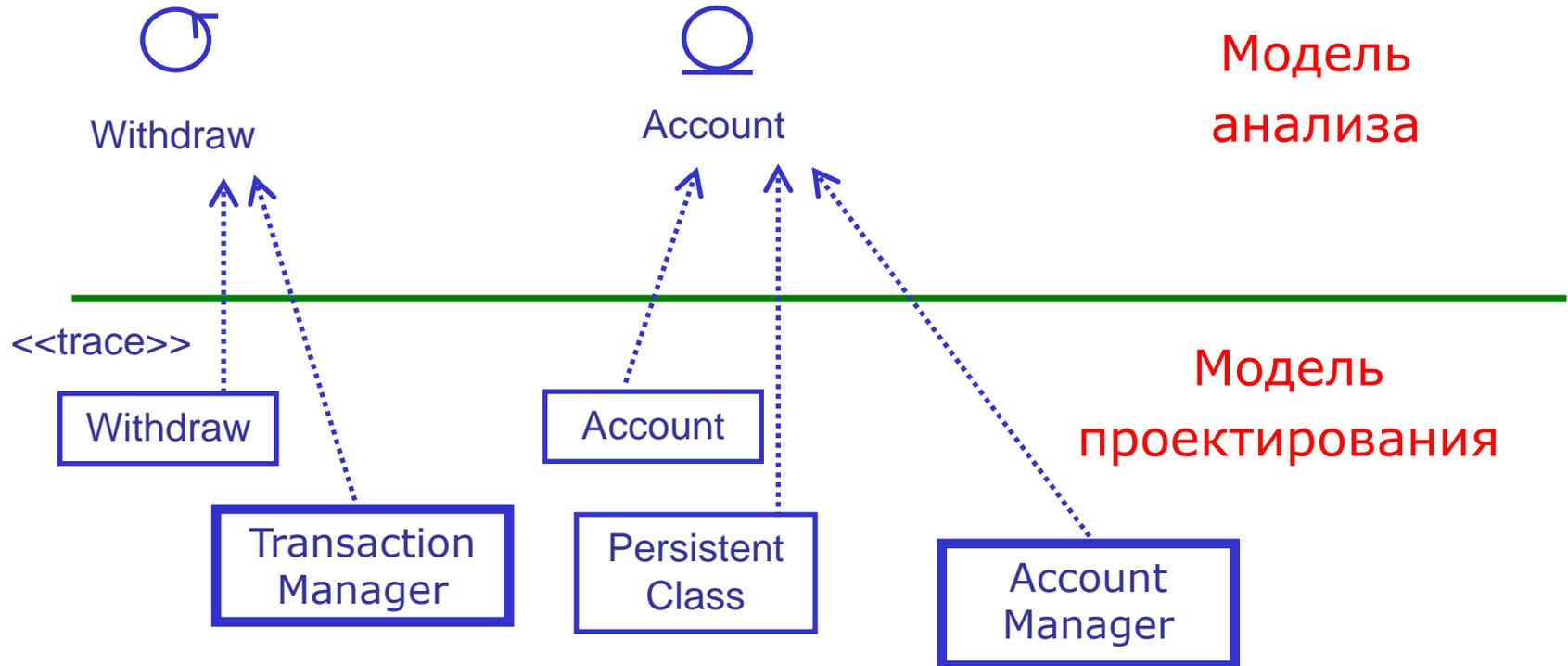


Построение модели проектирования.

Определение классов модели проектирования (2)

МГУ им. М.В.Ломоносова. Факультет ВМК.

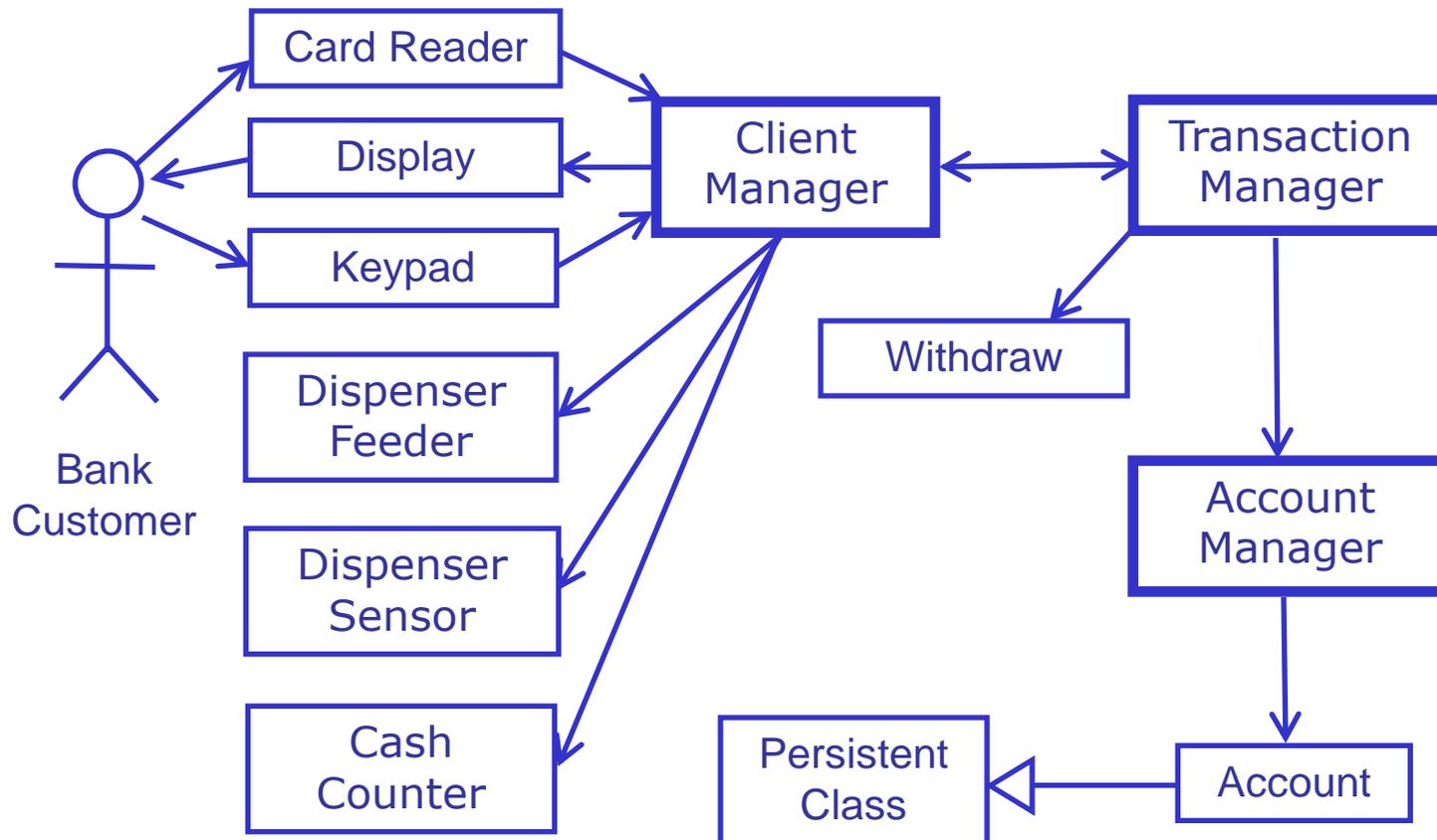
Романов Владимир Юрьевич ©2025



Построение модели проектирования. Определение отношений ассоциации

МГУ им. М.В.Ломоносова. Факультет ВМК.

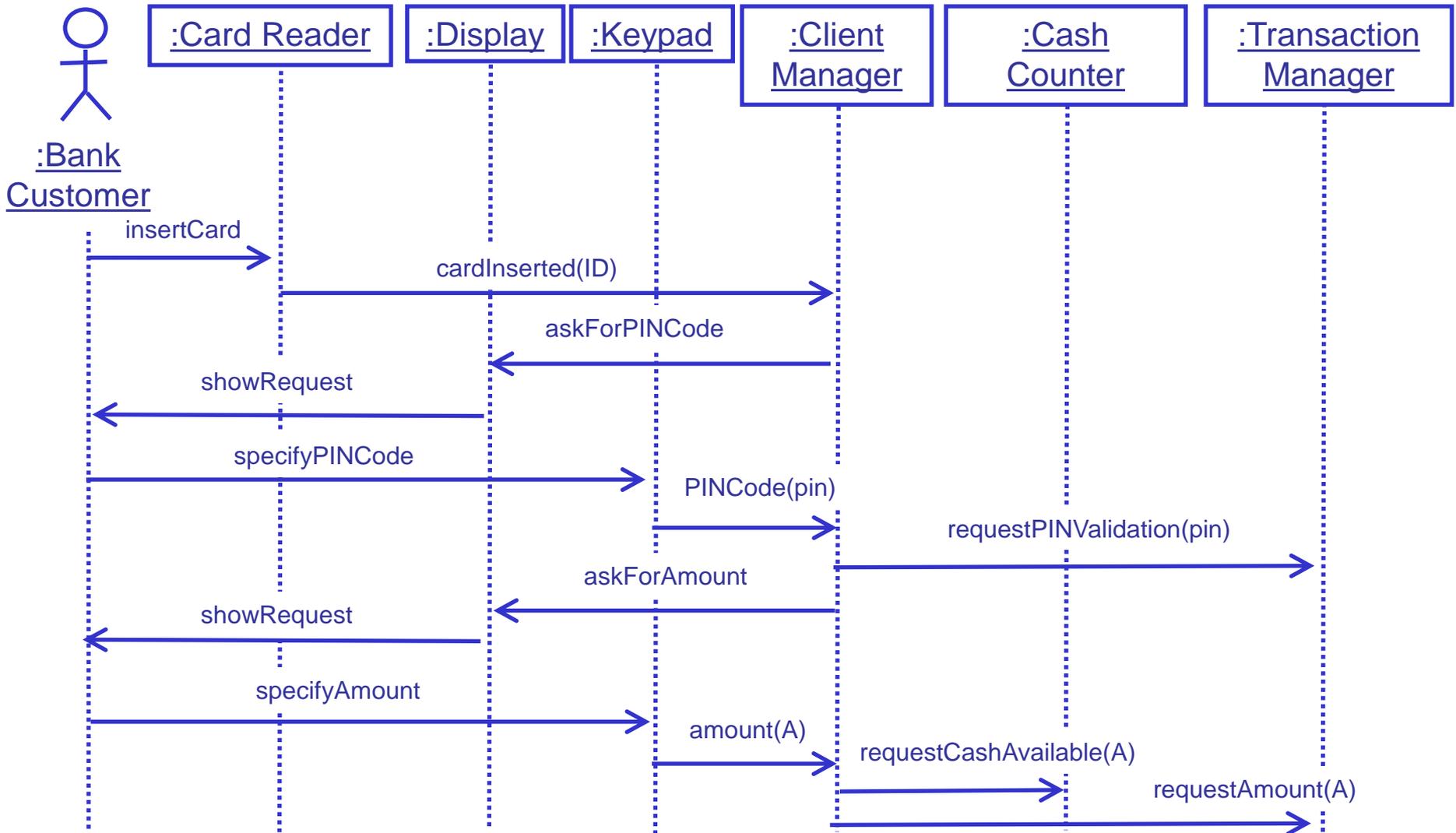
Романов Владимир Юрьевич ©2025



Построение модели проектирования. Описание реализации прецедента

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Построение модели проектирования. Пример

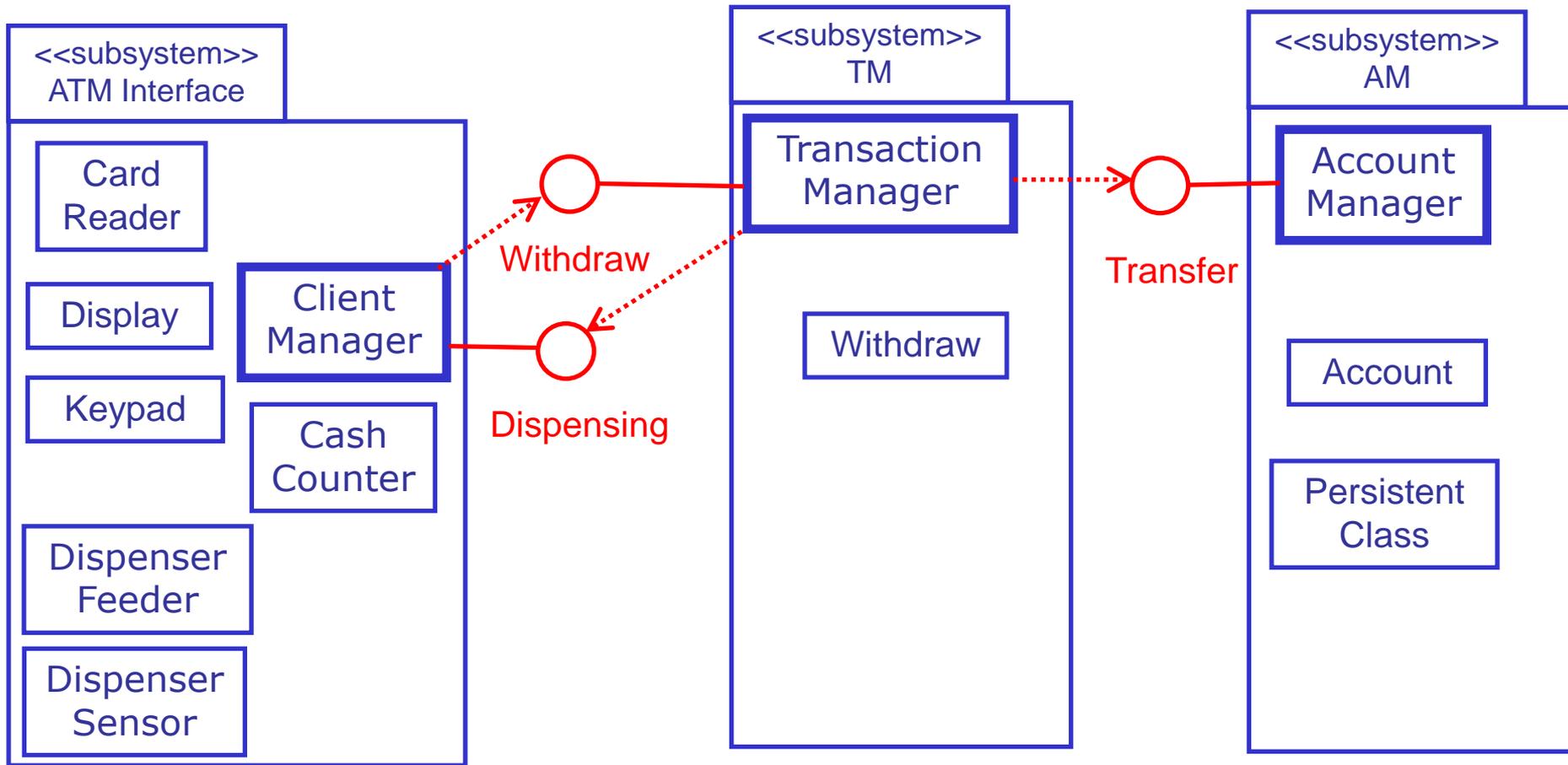
Системы с сотнями классов не управляемы =>

- 1. Классы группируются в подсистемы*
- 2. Подсистемы группируются в подсистемы*
- 3. Подсистемы нижнего уровня – **сервисные подсистемы***
- 4. В подсистемы группируются классы реализующие **необязательную (optional)** функциональность*
- 5. В подсистемы группируются **взаимосвязанные классы** (имеют тенденцию к общему изменению)*
- 6. Группирование на основе уже найденных классов (**снизу вверх**)*
- 7. Подсистемы и их интерфейсы определяет **архитектор (сверху вниз)***

Построение модели проектирования. Определение отношений ассоциации

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Построение модели реализации. Компоненты

1. Компонента

- физическая и заменяемая часть системы*
- реализует интерфейсы*

2. Для разработки исполняемой компоненты

- исходные файлы*
- таблицы баз данных*
- другие исполняемые компоненты (DLL или JAR)*

3. Сервисные компоненты

- для размещения на отдельном вычислительном узле*
- может быть несколько реализаций для каждого типа вычислительного узла*

Построение модели реализации из модели проектирования

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Модель проектирования



Модель реализации

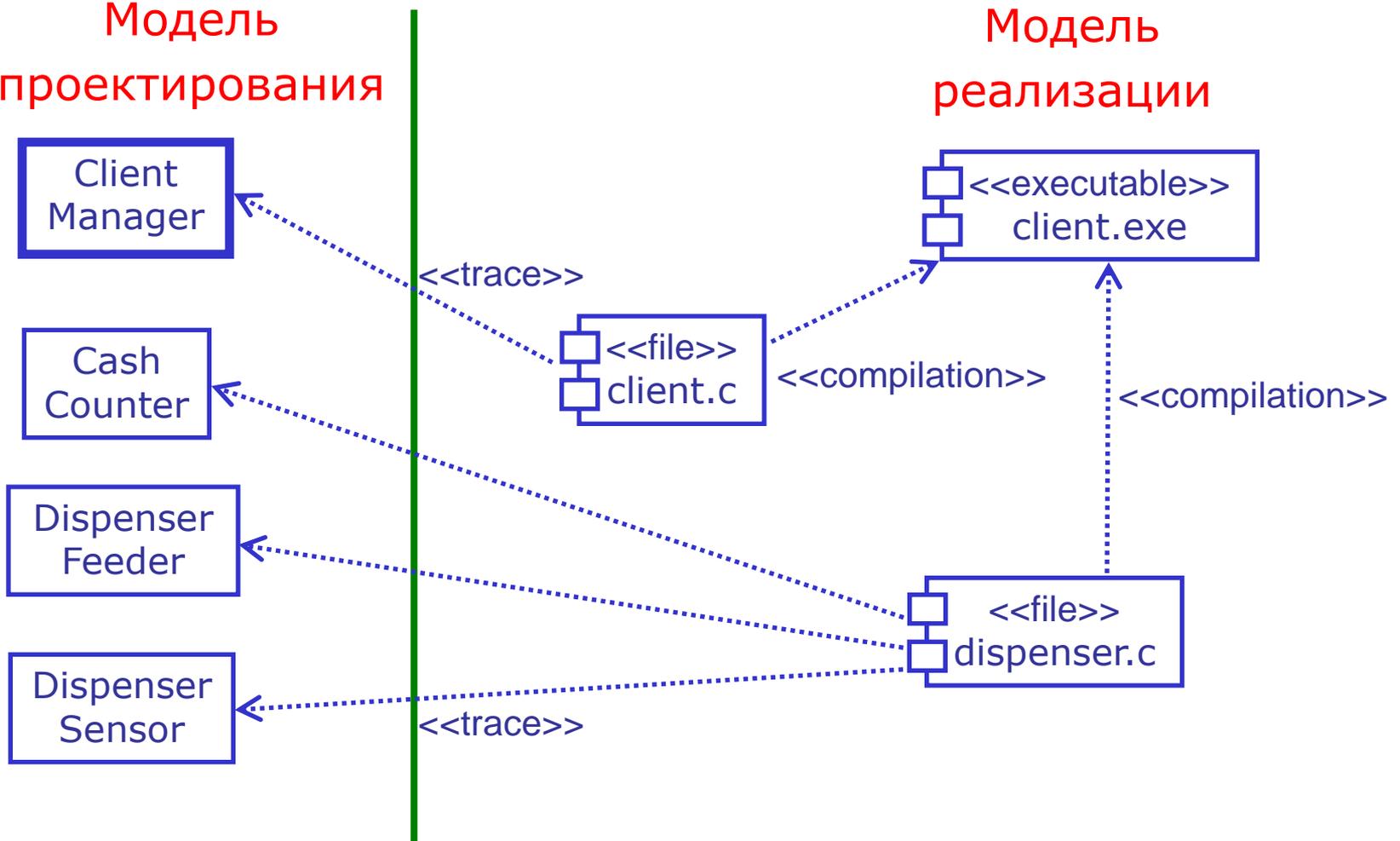


<<trace>>

<<compilation>>

<<compilation>>

<<trace>>



Построение модели тестирования. Прецеденты и процедуры тестирования

1. Прецедент тестирования (Test Case)

- Входные данные
- Условия тестирования
- Предполагаемые результаты

для

- конкретной реализации прецедента (черный ящик)
- конкретной ветви прохода через реализацию прецедента (белый ящик)
- конкретному не функциональному требованию

2. Процедура тестирования (Test Procedure)

описывает как:

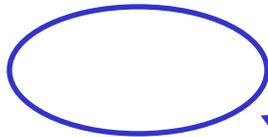
- выполнить инициализацию системы
- выполнить тест
- оценить результаты тестирования

Построение модели тестирования из модели прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

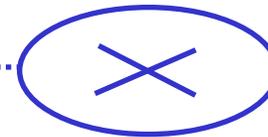
Романов Владимир Юрьевич ©2025

Модель
прецедентов

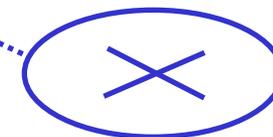


Withdraw
Money

Модель тестирования



Withdraw Money
Basic Flow



Withdraw Money
Without Cash

<<trace>>

<<trace>>

Построение модели тестирования. Пример теста

1. Ввод

- Клиент на счету 11-11-1111 имеет 350\$
- Клиент правильно идентифицировал
- Клиент запросил снять 200\$ со счета 11-11-1111
- денег в банкомате достаточно

2. Результат

- Баланс клиента уменьшился на 200\$
- Клиент получил 200\$

Отличительные особенности Унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Управляется прецедентами (use case driven)

 2. Ориентирован на раннюю разработку архитектуры

3. Итеративный и инкрементальный

Процесс ориентированный на разработку архитектуры системы.

Процесс ориентированный на разработку архитектуры системы (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Архитектура – общий взгляд на систему,
с которым согласны все разработчики системы

Архитектура – ясный взгляд на систему в целом,
дающий возможность управлять ее разработкой

Архитектура – описывает наиболее важные
элементы модели, которые

- управляют работой в каждой конкретной итерации
- управляют работой через весь жизненный цикл

Процесс ориентированный на разработку архитектуры системы (2)

Архитектура – содержит **некоторые** из:

- Подсистем
- Интерфейсов
- Вычислительных узлов
- Зависимостей
- Взаимодействий
- Активных классов

которые позволяют:

- Понять систему
- Разработать систему
- Оценить стоимость разработки системы

Процесс ориентированный на разработку архитектуры системы (3)

Архитектор проходит несколько итераций в *фазах*:

- *Inception* (начальная, анализа и планирования)
- *Elaboration* (исследование) цель фазы исследования – разработать архитектуры в виде **выполняемого фундамента** (*architecture baseline*)
- *Construction* (конструирование) – на фазе конструирования есть прочный фундамент для построения полной системы
- *Deployment* (внедрение)

Процесс ориентированный на разработку архитектуры системы (4)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Архитектор – специалист по интеграции частей проекта, но не эксперт по каждой части проекта.

Описание архитектуры используется разработчиками системы, но и них есть и более детальное описание

Архитектор дает взаимосвязанные *виды, проекции (view)* системы с помощью UML-диаграмм

Архитектор принимает решения о

- организации программной системы
- Структурных элементах и их интерфейсах, о взаимодействии элементов (*collaborations*)

Процесс ориентированный на разработку архитектуры системы (5)

- *Композиции элементов структуры и поведения в пакеты*
- *Структурных элементах и их интерфейсах, о взаимодействии элементов (collaborations)*
- *Описание архитектуры:
для каждой модели **4+1** вида*

Модель представления архитектуры 4+1

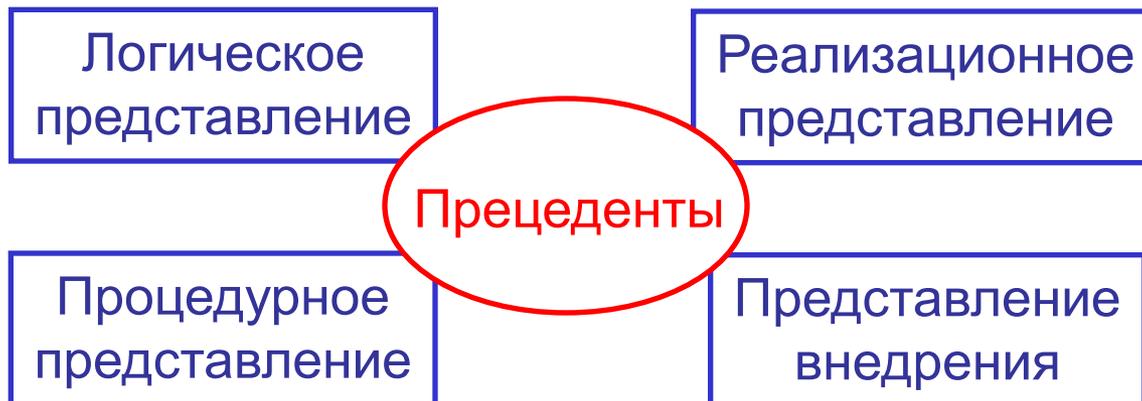
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Аналитики
Поведение системы

Конечный пользователь
Функциональные возможности

Программисты
Управление прогр.обеспечением



Системный интегратор
Функциональные возможности
Модульное наращивание
Производительность

Системотехника
Топология системы
Внедрение
Установка
Связи

Модель представления архитектуры

4+1

- **Прецеденты**
Ключевые сценарии и прецеденты. Используются для задания направления поиска других представлений
- **Логическое представление**
 - *Функциональные требования системы (что нужно пользователю)*
 - *Абстракция модели проектирования*
 - пакеты
 - подсистемы
 - классы

Модель представления архитектуры

4+1

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- **Реализационное представление**
Организация статических программных кодов
 - тексты
 - данные
 - компоненты
 - исполняемые коды

в терминах

- пакетов
- уровней
- управления конфигурациями

Модель представления архитектуры

4+1

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Процедурное представление*
 - *Параллельность*
 - задач
 - потоков
 - процессов
 - их взаимодействие
 - *Распределенность объектов*
 - *Отказоустойчивость*
- *Представление внедрения*
Распределение компонент на платформы и вычислительные узлы

Описание архитектуры.

- Сложность создания описания архитектуры – извлечение только самого важного без добавления в описание нового
- *Первая версия описания архитектуры появляется в конце фазы Elaboration*
- *Сложно сохранить описание системы небольшим*

Описание архитектуры. Зачем нужна архитектура

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- 1. Для понимания системы*
- 2. Для организации разработки системы*
- 3. Для стимулирования повторного использования уже написанного кода и компонент системы*
- 4. Для развития системы в дальнейшем*

Описание архитектуры. Понимание системы

- 1. Сложное поведение системы*
- 2. Работа в сложном окружении – сложно интегрировать во множество компонент разработанных сторонними разработчиками*
- 3. Используются сложные технологии (EJB или COM)*
- 4. Используют и индивидуальные пользователи и группы пользователей*
- 5. Разрабатываются распределенно (географически) и поэтому необходимо координировать работу*

Описание архитектуры. Организация разработки

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. *Организация разработки существенно упрощается, если может выполняться **распределенно** и **параллельно***
2. ***Назначение групп** ответственных за подсистемы дает возможность **распределенности** и **независимости***
3. ***Стабильные интерфейсы** – позволяют **независимо** развивать программу по обе стороны интерфейса*
4. *Правильная архитектура и **шаблоны проектирования** позволяют найти **правильные интерфейсы** между подсистемами*

Описание архитектуры.

Шаблоны архитектуры (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

***Facade** – предназначен для создания упрощенного интерфейса для группы подсистем или сложной подсистемы.*

Область применения:

- *Необходимо предоставить более простой вариант стандартного интерфейса*
- *Уменьшить зависимость клиента от подсистемы*
- *Нужно создать **слои** подсистемы*

Описание архитектуры.

Шаблоны архитектуры (2)

***Decorator** – предоставление механизма для добавления или удаления функциональности компонентов без изменения их внешнего представления или функции. Назначение:*

- *Необходимо динамически менять свойства классов
 - незаметно для пользователя класса
 - не испытывая ограничения механизма наследования классов*
- *Свойства компонента динамически включаются и отключаются*
- *Есть несколько независимых функций, применяемых:
 - динамически
 - в разной комбинации*

Описание архитектуры. Шаблоны архитектуры (2)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Layers – Способ организации модели проектирования на уровни.

Компоненты одного уровня могут ссылаться только на компоненты нижнего уровня

=> *Упрощение:*

- *Компоненты нижнего уровня не зависят от:*
 - *интерфейсов верхнего уровня*
 - *реализации верхнего уровня*

Описание архитектуры. Поощрение повторного использования

1. Архитектор может определить **повторно используемые подсистемы** тщательно проектируя их
2. Архитектор может определить **повторно используемые компоненты**
3. UML ускоряет создание повторно используемых подсистем и компонент
4. Должно предполагаться использование подсистем **вне текущего контекста**
5. Архитектор должен **решать более общую задачу**

Описание архитектуры. Развитие системы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- 1. Развитие стимулируется изменением окружения системы*
- 2. Может потребоваться новая функциональность системы*
- 3. Необходима устойчивость системы при ее изменении, а не ее деградация*

Описание архитектуры. Развитие системы. Принципы разработки системы (1)

1. *Функциональная модульность. Классы группируются в **optional сервисные подсистемы SSS**. SSS имеют только внутреннее сцепление (cohesion) => SSS независимы*

2. *Отделение проектирования интерфейсов от проектирования SSS*
 1. *=> несколько SSS могут поддерживать тот же интерфейс*

 2. *=> возможна замена подсистем, поскольку зависимость клиента только от интерфейса*

Описание архитектуры. Развитие системы. Принципы разработки системы (2)

3. *SSS – на этапе проектирования помещается в отдельную компоненту*

=> компоненты могут быть размещены на разные вычислительные узлы

4. *Необходимо уменьшение сцепления между SSS.*

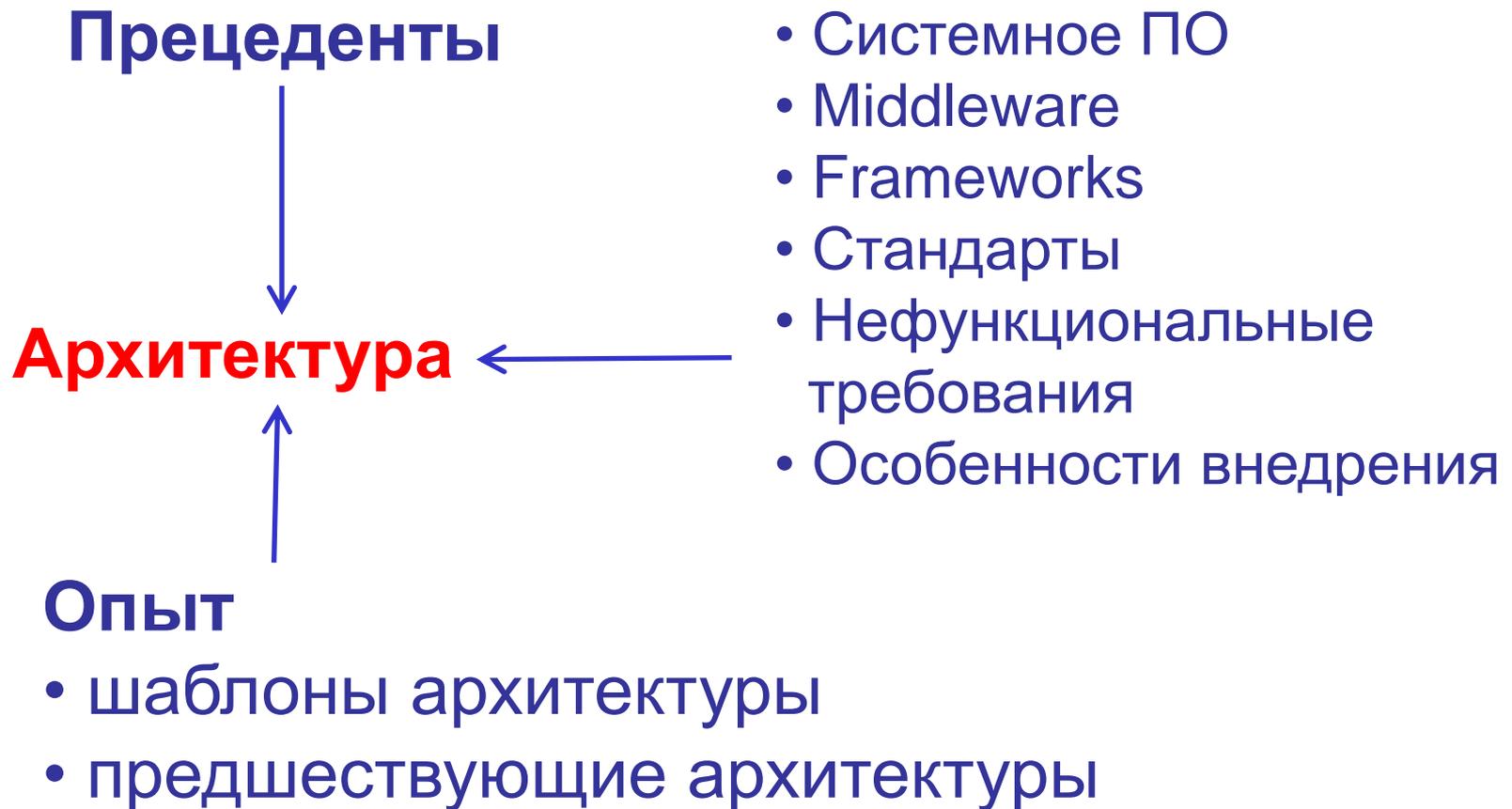
=> единственный способ общения между подсистемами через асинхронные сигналы

=> поощряется не только инкапсуляция, но и распределенность

Взаимосвязь прецедентов и архитектуры.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Шаги к архитектуре. (1)

- *Результат фазы Elaboration – фундамент архитектуры (скелет системы)*
- *Выбор прецедентов для построения архитектуры*
 - *те, которые позволяют оценить наиболее серьезные риски*
 - *Наиболее важная функциональность для пользователя*
- *Реализация, интеграция и тестирование фундамента архитектуры приводит к демонстрационной версии системы от которой идет обратная связь*

Шаги к архитектуре. (2)

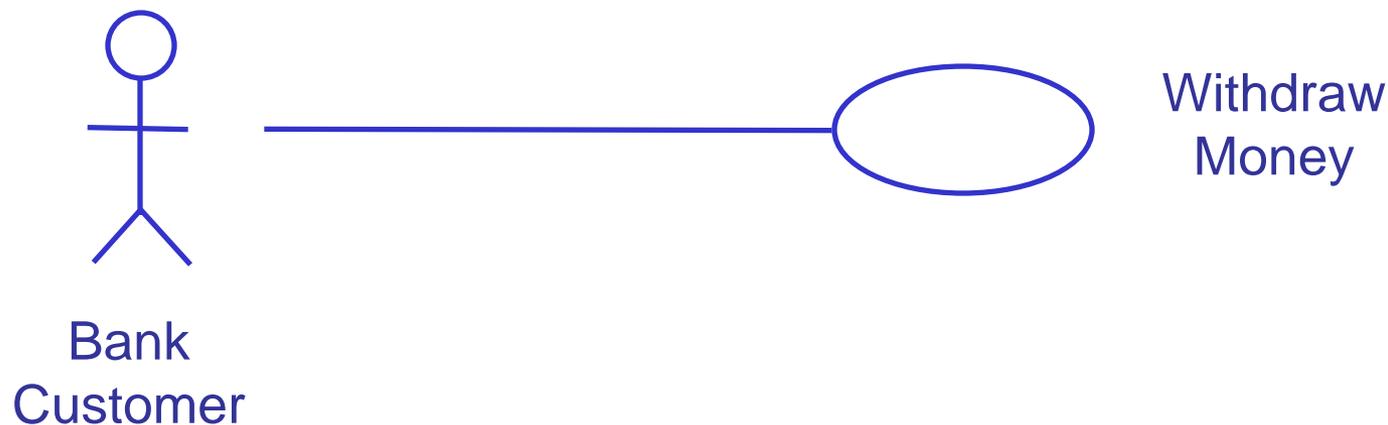
- *Фундамент архитектуры – небольшая тощая система которая имеет все модели которые имеет полная система в конце фазы конструирования*
- *Фундамент имеет скелет для:*
 - *Подсистем*
 - *Компонент*
 - *Вычислительных узлов*
Но и имеет также
 - *Поведение*
 - *Исполняемый код*
- *После **стабилизации** фундамента архитектуры заканчивается фаза уточнения*

Описание архитектуры. Модель прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *В модель прецедентов включаются наиболее важные актеры и прецеденты*
- *В нашей модели (условно) наиболее важным будем считать прецедент **WithdrawMoney***



Описание архитектуры. Модель проектирования

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Наиболее важные подсистемы*
- *Интерфейсы*
- *Несколько особенно важных классов*
- *Активные классы*
- *Реализация наиболее важных прецедентов с помощью описанных в архитектуре классификаторов*
- *Распределение активных классов по вычислительным узлам*

Модель проектирования (архитектура). Активные классы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Все **активные классы** включаются в описание модели проектирования архитектуры
- Классы необходимые для понимания реализации прецедента **WithdrawMoney**

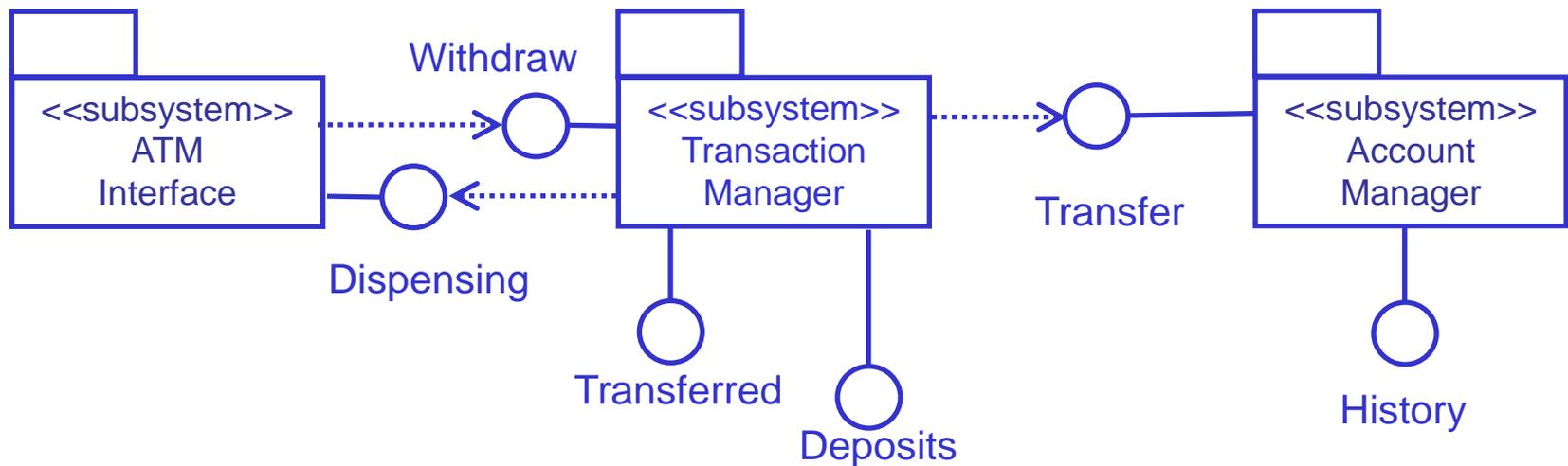


Модель проектирования (архитектура). Подсистемы и их интерфейсы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Подсистемы и интерфейсы необходимые для понимания реализации прецедента **WithdrawMoney***

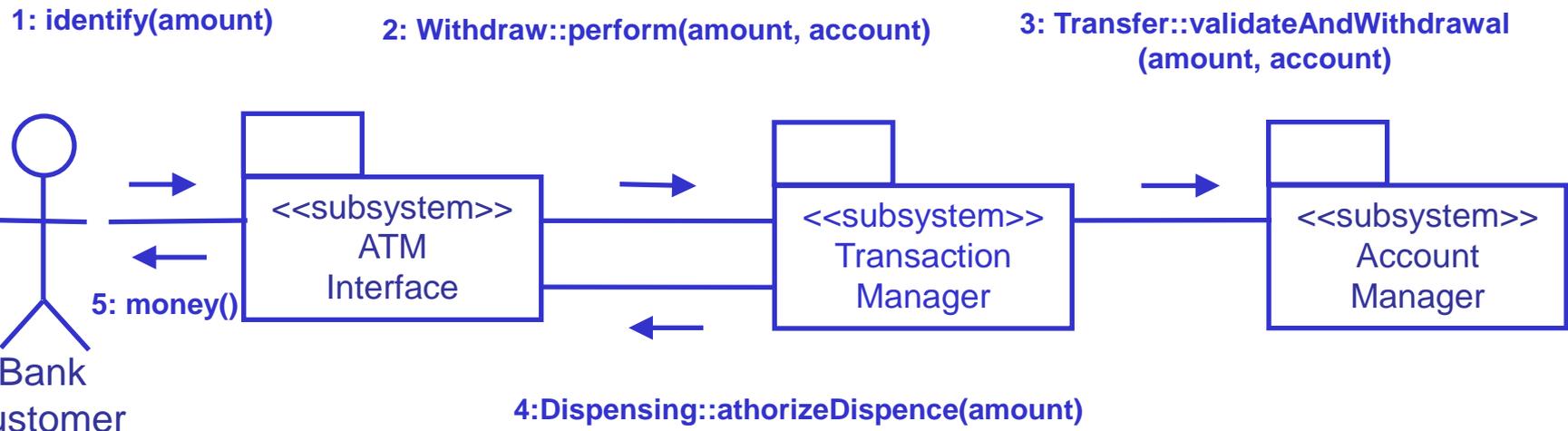


Модель проектирования архитектуры. Реализация прецедента WithdrawMoney

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Взаимодействие подсистем для реализации прецедента **WithdrawMoney***

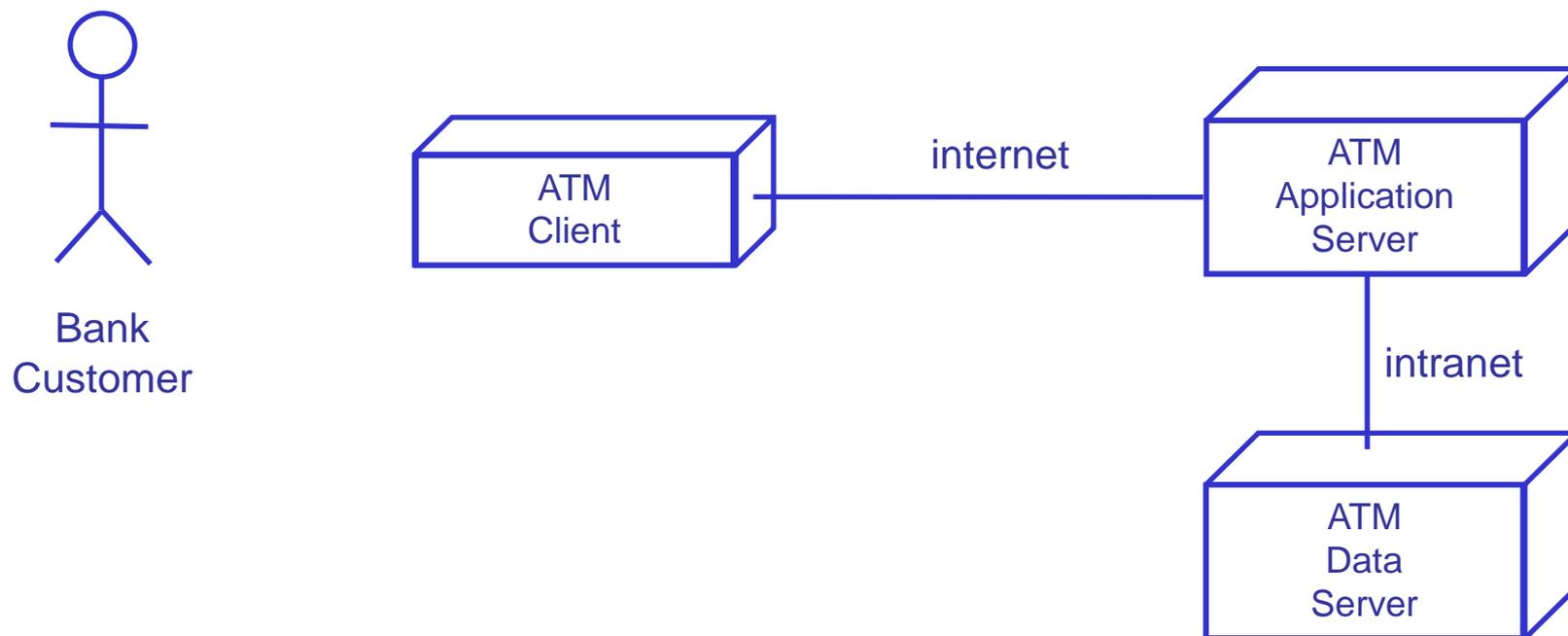


Модель внедрения (архитектура). (Deployment Model)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Классификаторы – вычислительные узлы*



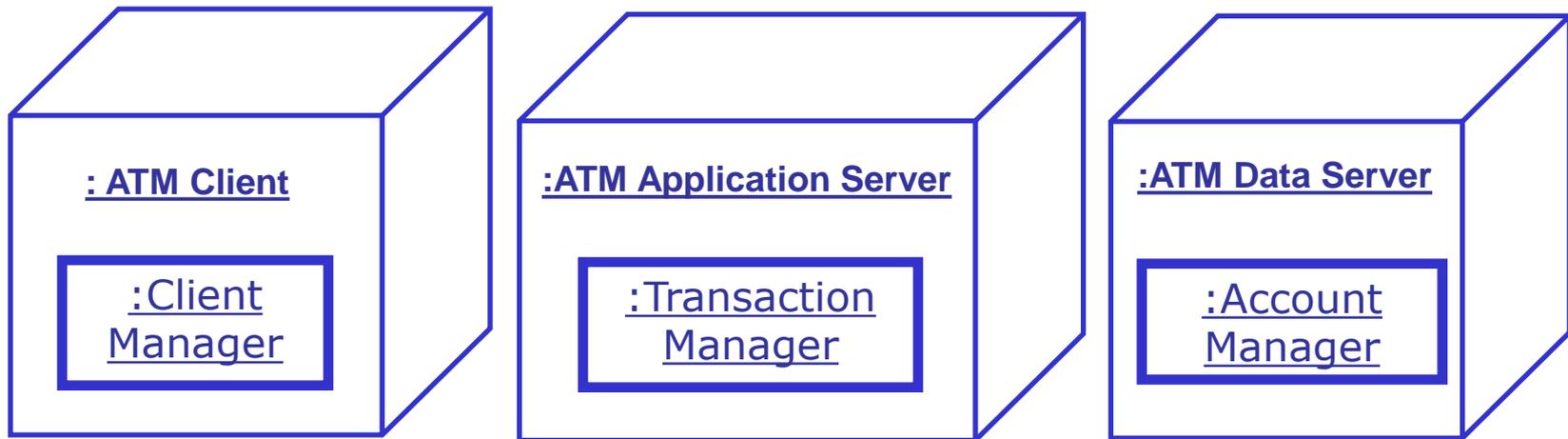
Модель внедрения (архитектура).

Активные объекты

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Активные объекты распределенные на экземпляры вычислительных узлов*



Модель внедрения (архитектура). (Deployment Model)

- *Определяется архитектура физической системы в терминах вычислительных узлов*
- *Модель внедрения может быть смоделирована на этапе сбора требований*
- *Для активных классов определяется:*
 - *Что делают активные объекты (экземпляры классов)*
 - *Жизненный цикл активных объектов: кто, когда и как создает и уничтожает активные объекты*
 - *Как активные объекты: общаются, синхронизируются, передают информацию*
- *Оцениваются возможности выч. узлов и их память*
- *Оценивается пропускная способность и доступность линий связи*

Отличительные особенности Унифицированного процесса

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Управляется прецедентами (use case driven)
2. Ориентирован на раннюю разработку архитектуры
-  3. Итеративный и инкрементальный

Итеративность и инкрементальность процесса.

Унифицированный процесс итеративный и инкрементальный (3)

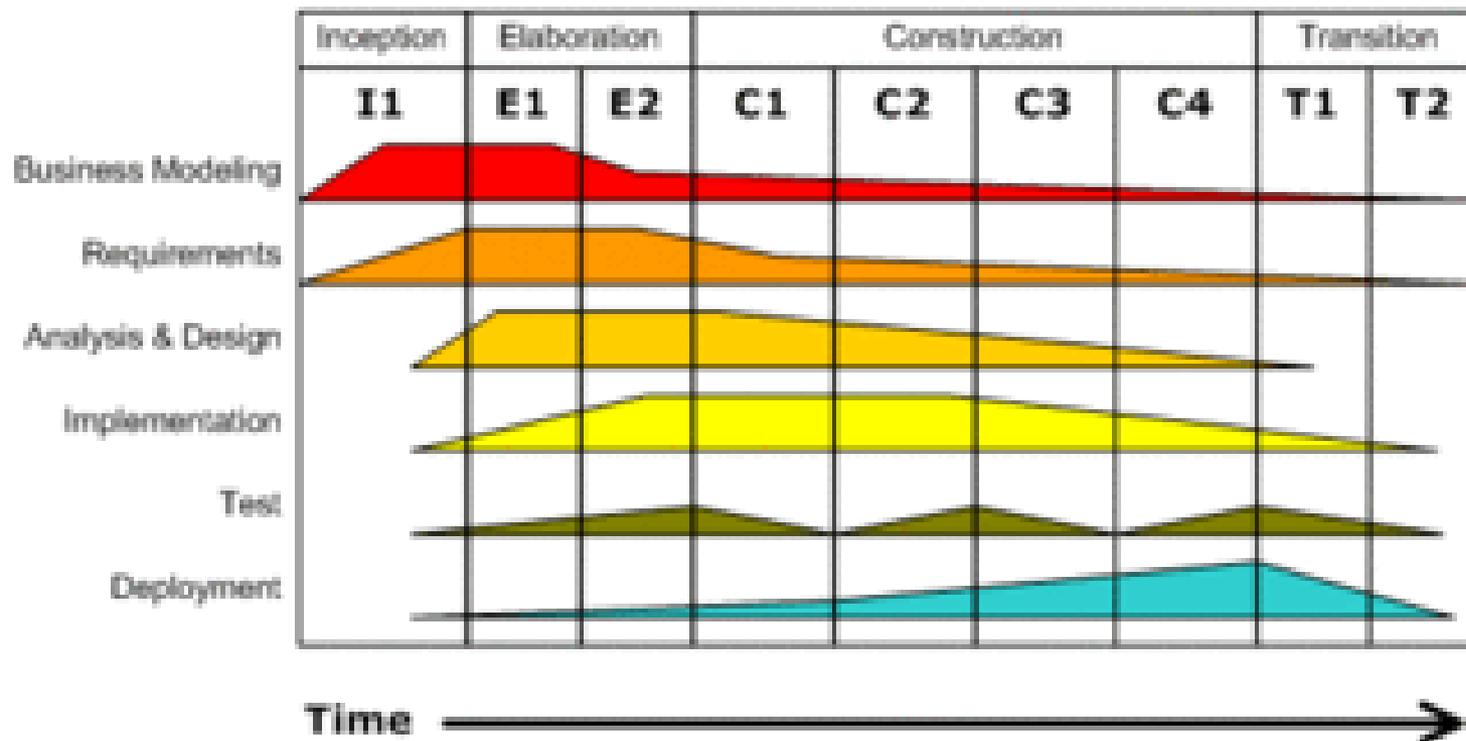
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Фазы итеративного процесса

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Итеративность и инкрементальность унифицированного процесса

- *Для эффективности унифицированного процесса необходимо иметь последовательность ясно сформулированных **контрольных точек (milestones)** которые дают команде **критерии окончания фазы** для перехода от одной фазы к другой фазе*
- *Внутри каждой фазы процесс проходит через итерации и приращения (increment)*

Фаза анализа и планирования требований

Фаза анализа и планирования требований (inception)

На этой фазе критерий – определение жизнеспособности проекта:

- 1. Определение и уменьшение рисков критичных для жизнеспособности системы*
- 2. Выделение ключевых требований –> моделирование прецедентов -> получение архитектуры-кандидата*
- 3. Получение оценки: стоимости, сроков разработки, качества разрабатываемого продукта*
- 4. Составление бизнес плана подтверждающего возможность выполнимость проекта*

Фаза проектирования

Фаза проектирования (Elaboration)

На этой фазе критерий – возможность построить систему

- 1. Определение и снижение рисков существенно влияющих на конструирование системы*
- 2. Определение большинства прецедентов, описывающих функциональность системы*
- 3. Переход от архитектуры-кандидата к исполняемому фундаменту архитектуры*
- 4. Подготовка детального плана проекта позволяющего выполнить фазу построения (construction)*
- 5. Уточнение предварительной оценки стоимости*
- 6. Законченный бизнес план – проект выгодно делать*

Фаза построения

Фаза построения (construction)

Критерий – способность системы начать работать в среде Пользователя

- *Достигается выполнением итераций и выпуском сборки (build) в конце каждой итерации*
- *В конце каждой сборки работоспособная функционально законченная версия системы*

Фаза построения

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Фаза построения (construction)

Критерий – способность системы начать работать в среде Пользователя

- *Достигается выполнением итераций и выпуском сборки (build) в конце каждой итерации*
- *В конце каждой сборки работоспособная функционально законченная версия системы*

Фаза внедрения

Фаза внедрения (transition)

Критерий – система готова к эксплуатации

- *Достигается модификацией системы в результате получения требований от пользователя не определенных на других фазах разработки*
- *Исправлением дефектов замеченных пользователем*

Итеративность и инкрементальность. Разработка небольшими шагами

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *У команды небольшие планы*
- *Спецификация, проектирование и реализация также небольшие*
- *Сборка, тестирование и исполнение кода в каждой итерации для каждого инкремента также небольшие*

При этом

- *Маленький шаг реализует функциональность необходимую пользователю*
- *Команда получает обратную связь от пользователя*
- *Снижаются риски вызывающие беспокойство*

Достоинства итеративной и инкрементальной разработки (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Ранняя обработка *критических и опасных* рисков
- Ускорение разработки *устойчивой архитектуры* с помощью которой будет управляться дальнейшая разработка
- Появляется скелет системы упрощающий сбор дополнительных требований к системе. Пользователю проще понять систему не в виде документации, а в виде прототипа
- Система разрабатывается постепенно, а не на последнем этапе разработки когда изменения в требованиях дороги. Пользователь проще оценить прогресс в разработке
- Процесс разработки строго регламентирован – увеличивается эффективность разработки

Достоинства итеративной и инкрементальной разработки (2)

- Ранняя обработка *критических* и *опасных* рисков
- Ускорение разработки *устойчивой архитектуры* с помощью которой будет управляться разработка
- Появляется скелет системы упрощающий сбор дополнительных требований к системе
- Система разрабатывается постепенно, а не на последнем этапе разработки когда изменения в требованиях дороги
- Процесс разработки строго регламентирован – увеличивается эффективность разработки

Достоинства итеративной и инкрементальной разработки (3)

- *Возможность раннего обучения*
 - *После нескольких итераций каждый в команде понимает смысл различных **потоков работ (workflows)***
 - *Легче обучать новых людей. Не требуется специальных «пилотных проектов» для новичков, которые затем выбрасываются*
 - *Ошибки новичков еще не критичны*
 - *Упрощается обучение новым технологиям. Риск их использования на ранних итерациях невелик => стимулируется использование этих технологий*

Достоинства итеративной и инкрементальной разработки (3)

- *Возможность раннего обучения*
 - *После нескольких итераций каждый в команде понимает смысл различных **потоков работ (workflows)***
 - *Легче обучать новых людей. Не требуется специальных «пилотных проектов» для новичков, которые затем выбрасываются*
 - *Ошибки новичков **еще** не критичны*
 - *Упрощается обучение новым технологиям. Риск их использования на ранних итерациях невелик => стимулируется использование этих технологий*

Итеративный процесс управления рисками. Риски представления системы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Риск – опасность для разработки, степень которой еще неизвестна

Риски:

- *Идентифицируются*
- *Получают приоритеты*
- *Отслеживаются*

Примеры рисков представления системы

- *Скорость работы*
- *Память используемая системой*
- *Точность вычислений*
- *Надежность*
- *Доступность*

Итеративный процесс управления рисками. Организационные (не технические) риски

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Организация не имеет людей опытных в уникальных областях необходимых для построения системы*
- *Предлагается новый язык реализации, но не все его знают*
- *Заказчик требует слишком быстрого появления системы*
- *Необходимо включать в разработку внешние фирмы*

Итеративный процесс управления рисками. Работа с рисками

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

После идентификации рисков и задания их приоритетов

- *Уход от риска, перепроектировав проект или изменив требования к системе*
- *Ограничить риск, что бы при этом затрагивалась по возможности меньшая часть системы или проекта*
- *Материализация риска, проверив оправдались ли опасения. Затем уход от риска или ограничение риска*
- *Риск оправдался. Если исправить невозможно, то возможен отказ о проекта с минимальными затратами времени и денег, до вовлечения в проект большого числа людей*

Типовая итерация

- *Итерация* – мини-проект при полном проходе по всем *потокам работ (workflows)*
- *Поток работ* - кооперация между *исполнителями (workers)* производящими и использующими *артефакты*
- *Пять основных потоков работ:*
 - *Определение требований*
 - *Анализ*
 - *Проектирование*
 - *Реализация*
 - *Тестирование*

Отличие итеративного процесса от модели водопада (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Между итерациям существует *перекрытие*

Итерация1



Итерация2



Итерация3



Распределение сбора артефактов по различным фазам унифицированного процесса

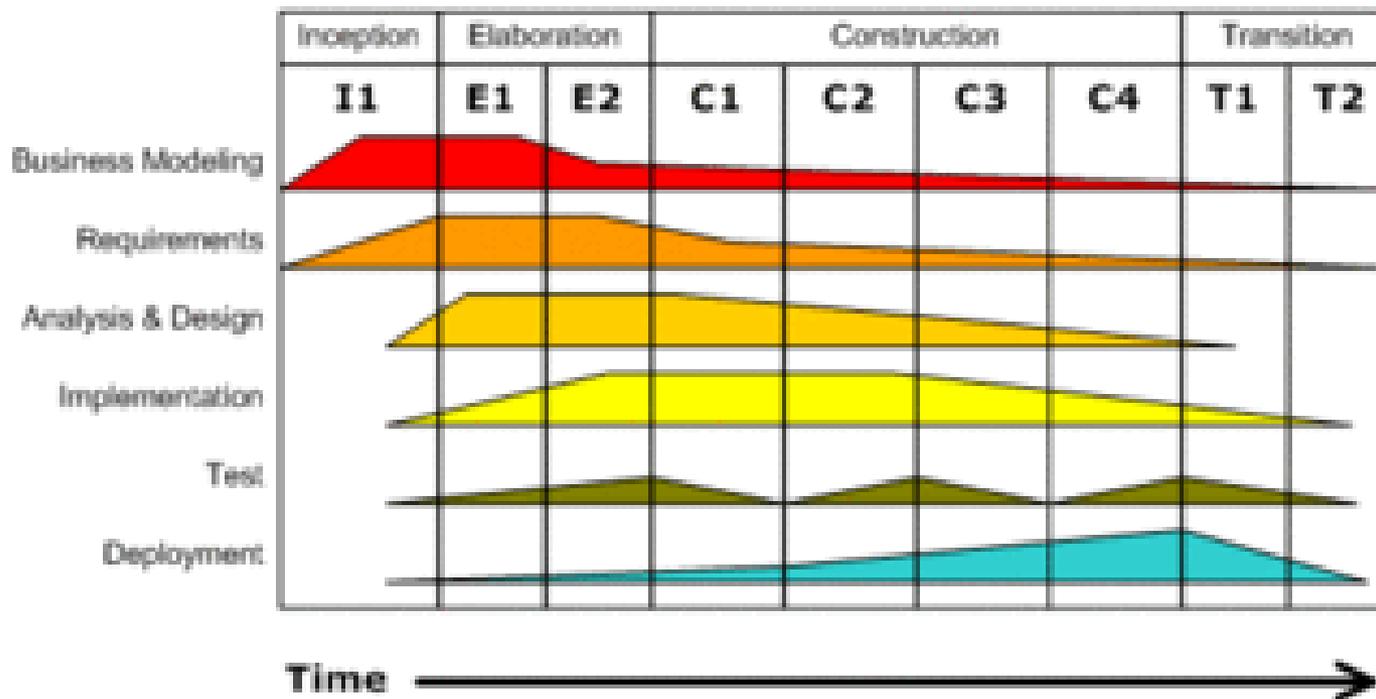
МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Сдвиг важности каждого потока работ в течении итераций и фаз процесса разработки

Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Проверка команды с помощью итеративного подхода

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- *Тенденция переходить к написанию кода.
Строки кода – то, что менеджеры способны сосчитать*
- *Сопrotивление изменениям
Изменения замедляют написание кода*
- *Нет заинтересованности в многократном использовании анализа, проекта или кода
Возможно потому что работу оценивают по написанию нового кода*

Изменение

- *Теперь основное:*
 - *Уменьшение рисков*
 - *Создание фундамента архитектуры*
 - *Реализация функциональности*
- *Прогресс изменяется в рисках, прецедентах, и компонентах реализующих прецедент*

Последствия итеративного подхода

- *Для создание бизнес-плана в фазах анализа и определения требований, необходимо снижать риски и создание демонстрационной версии системы*
- *Для создания законченного бизнес плана в конце фазы Elaboration организация должна быть уверена что не осталось невыясненных рисков и спланировать дальнейшие действия исходя из существующего фундамента архитектуры и требований к системе*
- *Для минимизации затрат организация должна использовать многократно используемые компоненты полученные в результате ранней разработки архитектуры*

Последствия итеративного подхода

- *Для избегания задержки разработки и роста цены разработки организация должна делать сначала сложные вещи*
- *Для избегания устаревания продукта к моменту выхода на рынок организация должна соглашаться на внесение изменений в систему.*

Разбитый на фазы Итеративный подход позволяет вносить изменения в течение всего времени разработки системы

Типовая итерация

- *Исполнители и артефакты могут участвовать более чем в одном потоке работ.*

Например, разработчик компонент (component engineer) участвует в потоках работ по анализу, проектированию и реализации

- *После завершения итерации*
 - *Регрессионное тестирование*
 - *Оценка изменений требований*
 - *Оценка появления новых требований*

Поток работ для получения требований к
системе как сценариев использования
системы.

Сбор требований к системе как построение модели прецедентов использования системы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

- Основная задача потока работ по определению требований – построение модели *прецедентов использования* системы
- Функциональные требования структурируются на прецеденты использования
- Нефункциональные требования привязываются к конкретным прецедентам использования системы
- Общие для нескольких прецедентов нефункциональные требования выделяются в отдельный документ называемый *дополнительные требования (supplementary requirements)*

Построение модели прецедентов

- Построение модели прецедентов требует от **системного аналитика** привязывать функциональность системы к конкретному пользователю системы – **актеру** (*actor*)
- Требует от системного аналитика выявлять категории *пользователей* системы - *актеров*
- Какие задачи (*прецеденты использования*) должна выполнить система для конкретных *категорий пользователей*
- Затем построенная **модель прецедентов** управляет всей разработкой

Построение модели прецедентов

Схема описания потока работ

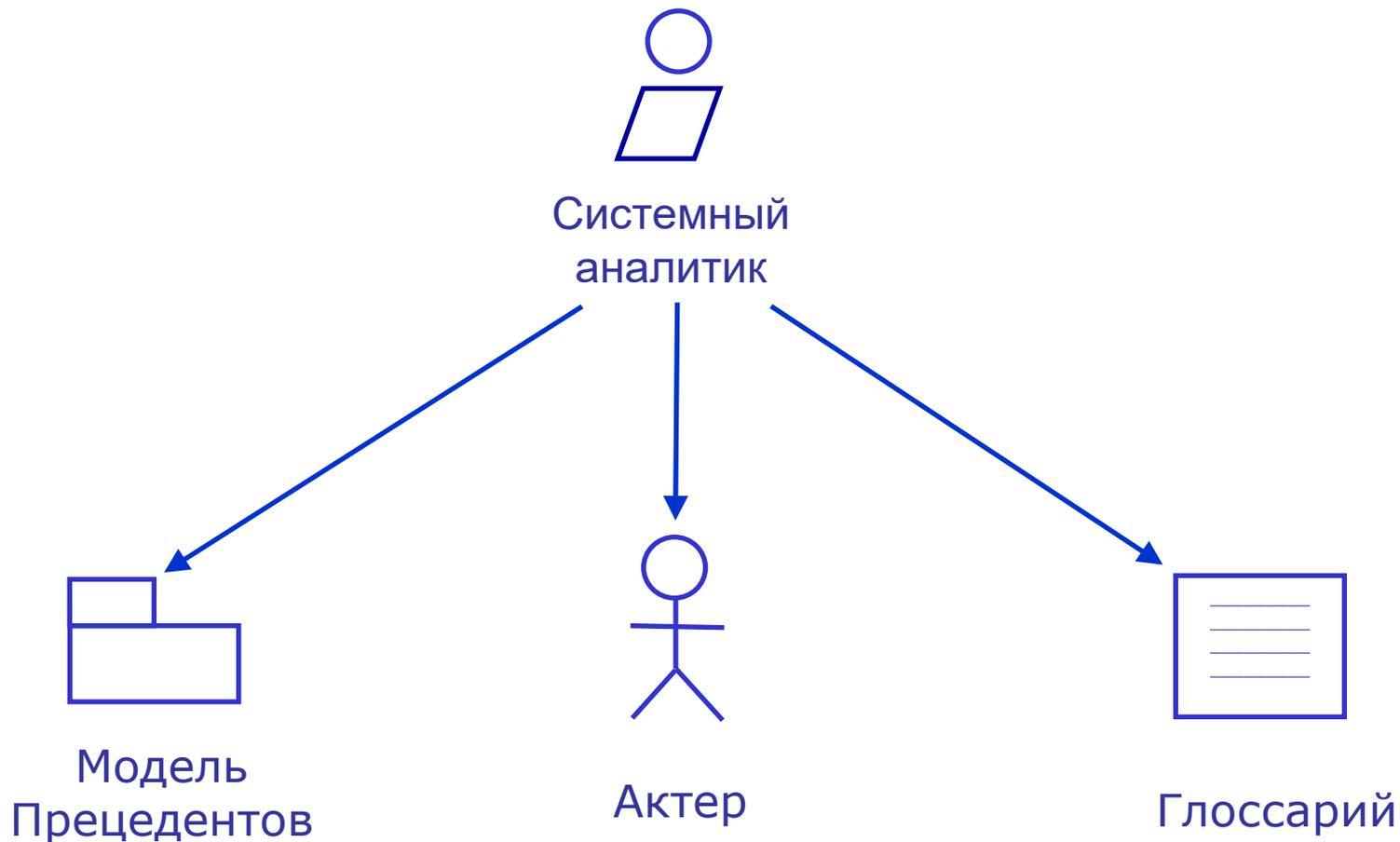
1. Описание **артефакта (artifact)**. Артефакт – общий термин для любого вида информации (создаваемой, изменяемой, используемой) **исполнителями (workers)** при работе по созданию системы
2. Описание **исполнителя (worker)** – сотрудника фирмы, участник потока работ. Обозначение роли которая может быть поручена одному человеку или группе лиц имеющих для этого соответствующие навыки
3. Описание **деятельности (activity)** – часть работы за выполнение которой отвечает **исполнитель** в потоке работ и имеет результат в виде набора **артефактов**

Построение модели прецедентов

Исполнители и артефакты (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

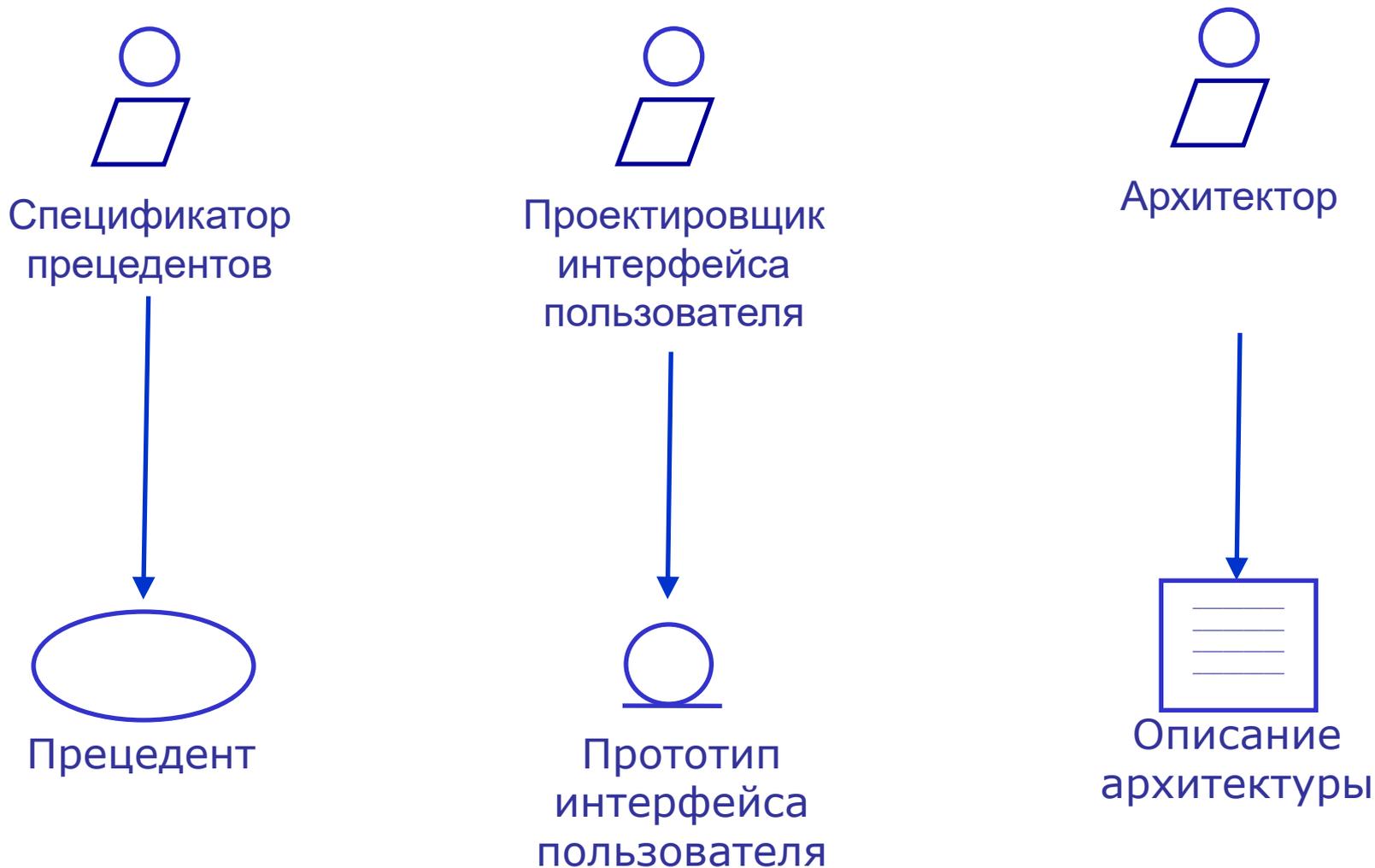


Построение модели прецедентов

Исполнители и артефакты (2)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Артефакты в потоке работ по построению модели прецедентов (1)

Модель прецедентов

- Позволяет разработчикам системы договориться с заказчиками и пользователями системы о требованиях к разрабатываемой системе
- Позволяет заключить соглашение (контракт) на разработку системы
- Структурировать требования на пакеты содержащие прецеденты и диаграммы прецедентов
- Модель прецедентов - исходная модель для построения моделей анализа, проектирования, реализации, тестирования

Артефакты в потоке работ по построению модели прецедентов (2)

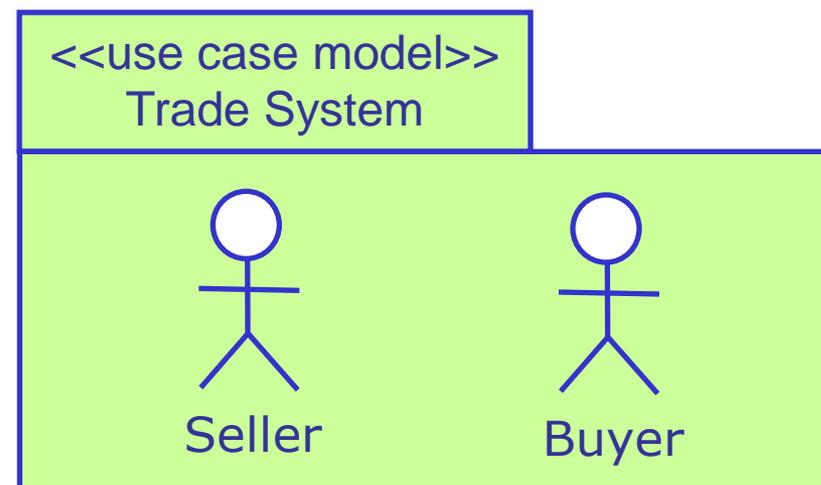
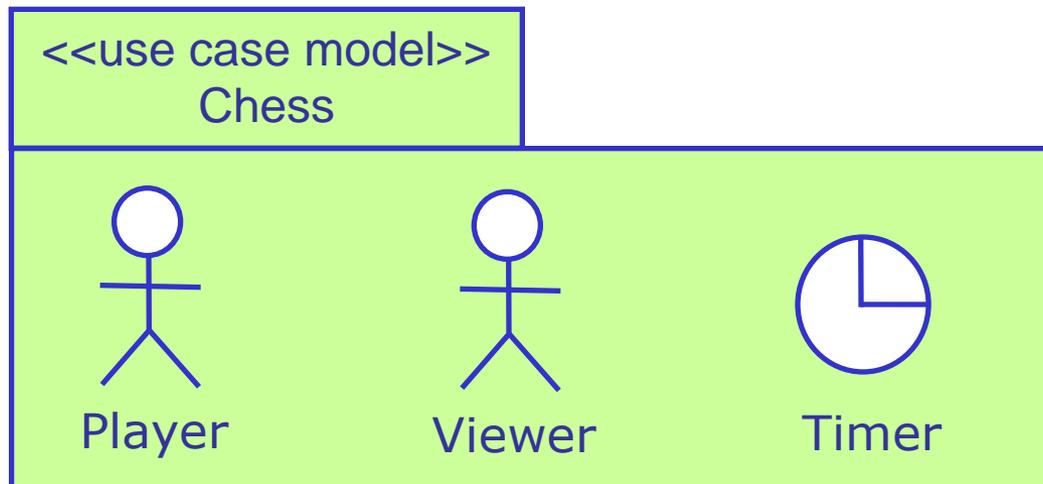
Актер

- Модель прецедентов описывает действия системы для каждой *категории пользователей – актеров*
- Актером может быть внешняя программная система
- Актером может быть внешнее устройство: часы, телефонная карта, ...
- Актерами могут быть сотрудники или клиенты фирмы для которой разрабатывается система.
И сотрудники, и клиенты могут быть взяты из **бизнес модели**

Построение модели прецедентов. Примеры артефактов - актеров

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

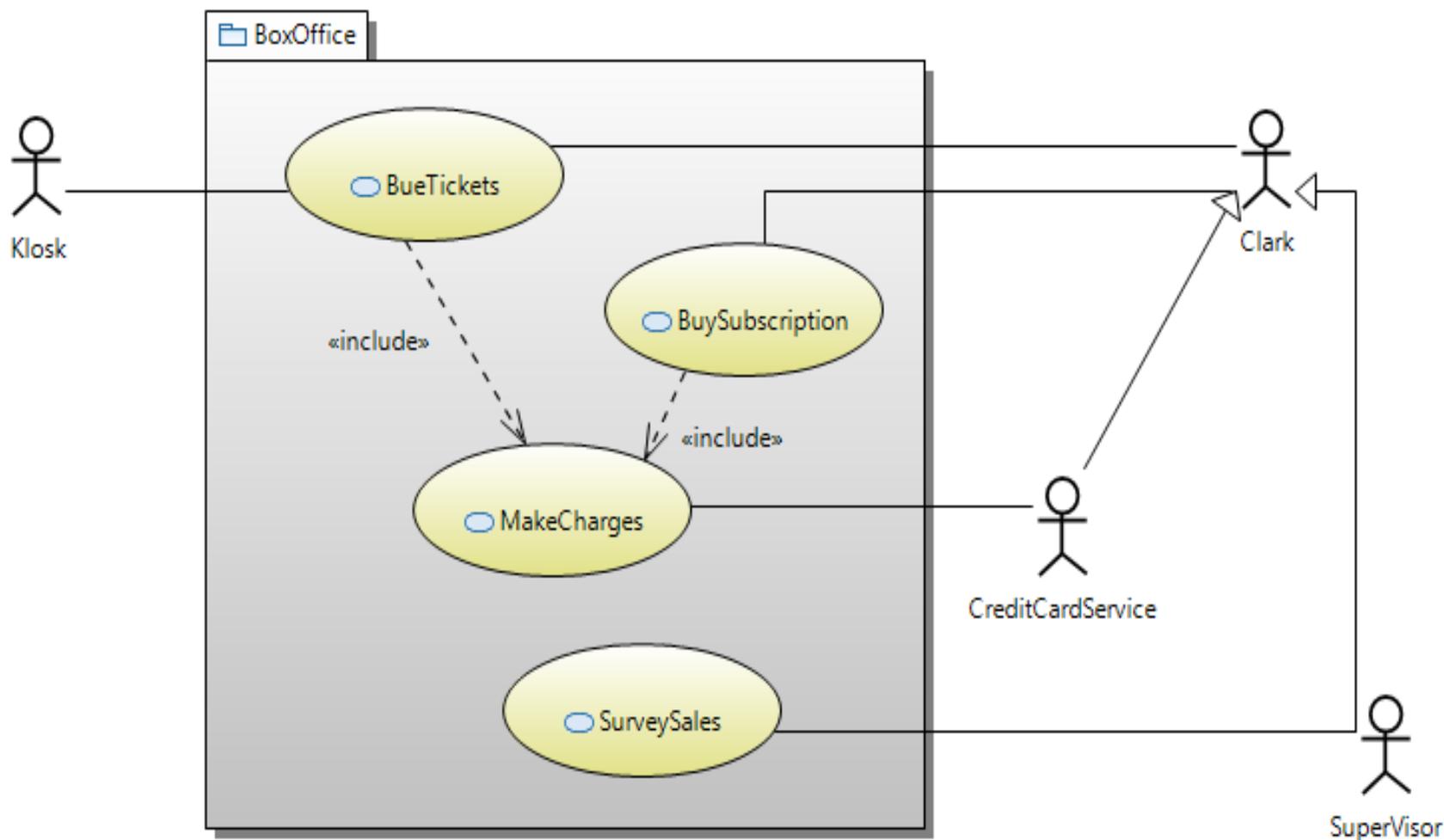


Построение модели прецедентов.

Примеры артефактов – актеров и прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

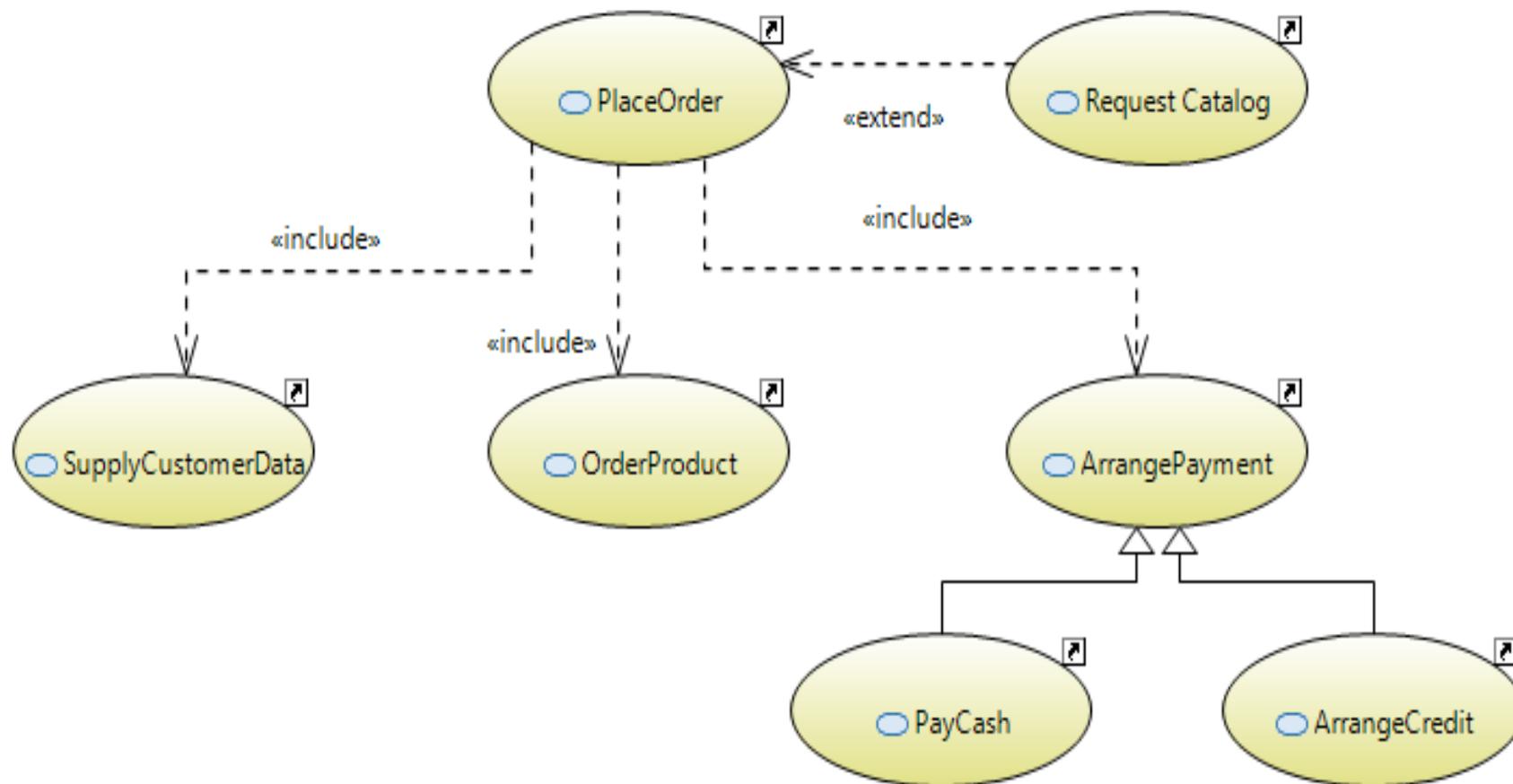


Построение модели прецедентов.

Примеры отношений прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

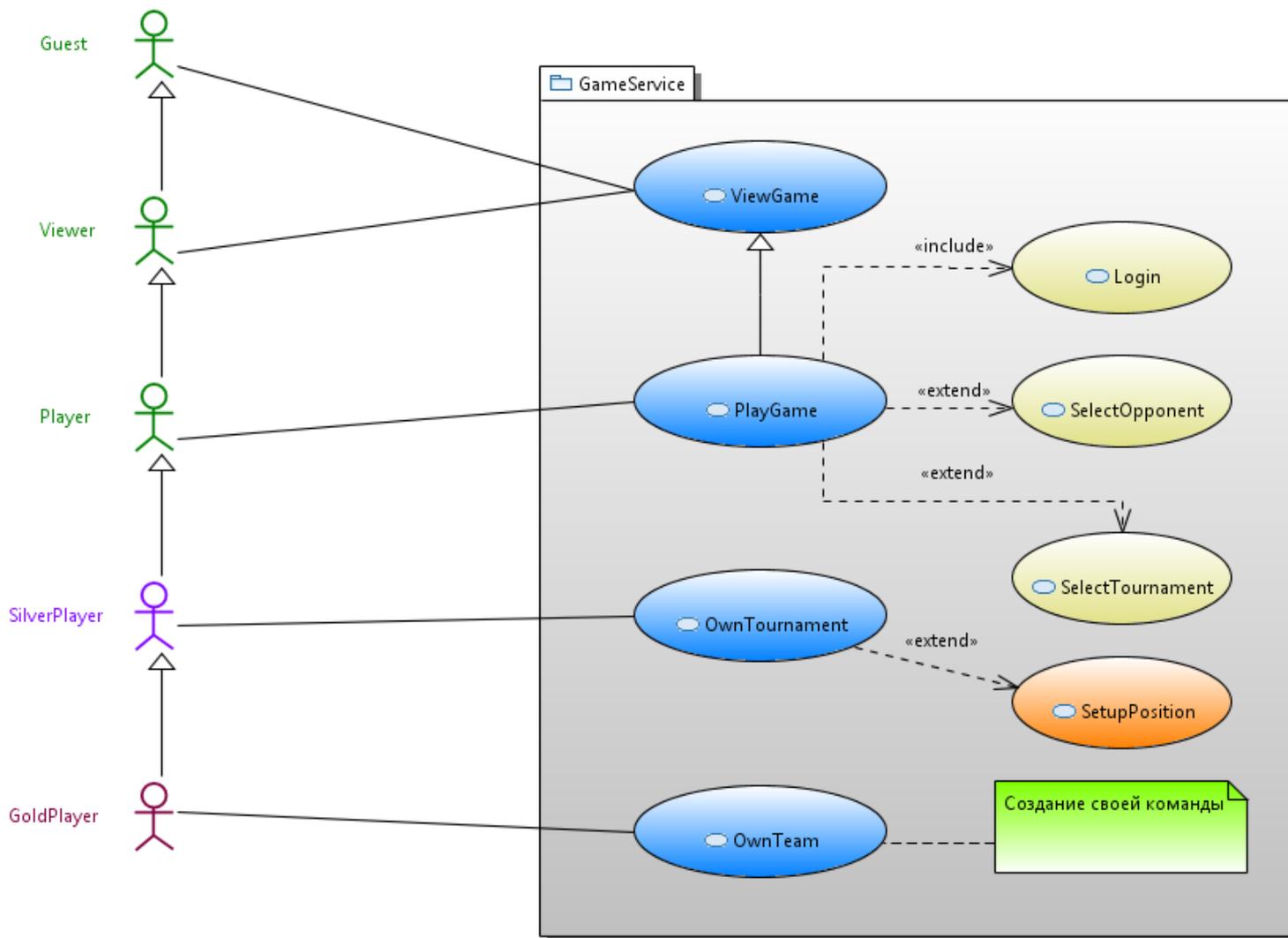
Романов Владимир Юрьевич ©2025



Построение модели прецедентов. Сервис настольных игр

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Артефакты в потоке работ по построению модели прецедентов (1)

Прецедент

- Прецедент – способ использования системы.
- Описывает последовательность действий с альтернативами, которые система может выполнить взаимодействуя с актерами
- Прецедент – классификатор для описания поведения => имеет атрибуты и операции
- Прецедент может включать :
 - диаграммы переходов и состояний,
 - диаграммы деятельности,
 - диаграммы взаимодействия
 - диаграммы последовательности взаимодействия

Артефакты в потоке работ по построению модели прецедентов. Прецедент (2)

Прецедент

- *State Transition Diagram* – описывает жизненный цикл прецедентов в терминах состояний и переходов
- *Activity Diagram* – более подробное описание, описывая деятельности выполняемые при переходе на диаграмм состояний и переходов
- *Collaboration Diagram* и *Sequence Diagram* – используются для описания взаимодействия между типовым экземпляром актера и типовым экземпляром прецедента

Артефакты в потоке работ по построению модели прецедентов. Прецедент (3)

Типовой жизненный цикл прецедента

- Инициализация прецедента и установка прецедента в начальное состояние
- Активизация прецедента внешним сообщением
- Переход в другое состояние, с выполнением некоторой последовательности действий. Эта последовательность может содержать вычисления, выбор пути на диаграмме и посылку сообщения актеру
- Ждет в новом состоянии прихода нового сообщения от актера
- Активизация новым сообщением и повторение цикла.

Артефакты в потоке работ по построению модели прецедентов. Прецедент (3)

Особенности прецедента

- Атрибуты прецедента локальны для экземпляра прецедента (не могут быть использованы другими экземплярами прецедентов). Взаимодействие прецедента только с актерами.
- Нет конфликта между прецедентами из-за данных.
- Не требуется синхронизация с другими прецедентами
- Прецедент выполняется полностью или не выполняется совсем.
- Цель – сделать описание простым, что бы модель можно было обсуждать с пользователем и заказчиком.

Артефакты в потоке работ по построению модели прецедентов. Описание архитектуры

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Описание архитектуры (модель прецедентов)

- Важная и критичная функциональность
- Важные требования, которые должны быть реализованы в начале жизненного цикла разработки системы
- Описание архитектуры используется для задания приоритетов в реализации прецедентов.

Использование будет описано при рассмотрении *деятельностей* потока работ по построению модели прецедентов

Артефакты в потоке работ по построению модели прецедентов. Глоссарий

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Глоссарий

- Определение важных или общих для всего проекта понятий
- Важен для получения согласованных определений приемлемых пользователями, заказчиками, аналитиками и разработчиками системы
- Может возникать из бизнес модели

Артефакты в потоке работ по построению модели прецедентов. Прототип интерфейса

Прототип интерфейса пользователя

- Прототип интерфейса помогает сбору требований к системе
- Позволяет построить лучший интерфейс системы, с которой пользователь будет работать основную часть рабочего времени
- Способствует лучшему пониманию взаимодействия актеров и прецедентов при построении модели прецедентов
- При построении прототипа интерфейса могут использоваться артефакты: модели интерфейса пользователя и скриншоты

Построение модели прецедентов. Прототип интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Latvian gambit

Директор турнира:   Leo_C  

Присоединиться

Контроль времени	3 дня	Игроки	14/60
Рейтинговые партии	Да	Средний рейтинг	1605
Диапазон рейтинга	> 1400	Баллов доступно	539
Макс. размер группы	6	Завершено	0%
# В следующий этап	2	раунд	0
Партий одновременно	10	Завершенные партии	0
КБ	Да	Оставшиеся партии	0 (текущий раунд)
Макс. ср. время на ход	∞	Макс. поражений по времени %	20%

The latvian gambit is a fascinating opening. Although it is not entirely sound it creates a very fun game!

Построение модели прецедентов. Прототип интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

 Игроки (14)



Altotemmi 🇮🇹 👑
John Temming
(1756)



Ziggy_Zugzwang 🇩🇰 ★
Charlie Dorridge
(1772)



LttlePrince 🇵🇱
Marcin Borchardt
(1879)



Frank9373 🇺🇸
Frank
(1833)



Raf89 🇫🇷 ★
(1668)



Trompicon 🇮🇹 ★
Pablo Daniel Bebebino
(1723)



aprilis 🇮🇹
Laszlo
(1589)



MIRLOIII 🇪🇸 💎
Francisco Garcia Lechugo
(1463)



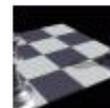
statundu 🇮🇩
Ali Suwarjono Tatundu
(985)



Invisibilia 🇨🇪
(1480)



Leo1706 🇵🇰
Yasir Shabbir
(1468)



RogerJim 🇮🇹
(1592)

Построение модели прецедентов. Прототип интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

<https://www.chess.com/club/matches/team-russia/908450>

<https://www.chess.com/club/matches/917504>

Исполнители в потоке работ по построению модели прецедентов.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Исполнитель (*worker*) – абстракция человека с некоторыми способностями и знаниями

Это не конкретный человек (один человек – много исполнителей)

Это не должность. Должность одна – исполнителей много

Для исполнителя выполняется:

- Описание ответственности исполнителя
- Описание предполагаемого поведения исполнителя

Системный аналитик (System analyst)

Спецификатор прецедентов (Use Case Specifier)

Проектировщик интерфейса пользователя (UI Designer)

Исполнители в потоке работ по построению модели прецедентов. Системный аналитик

Системный аналитик

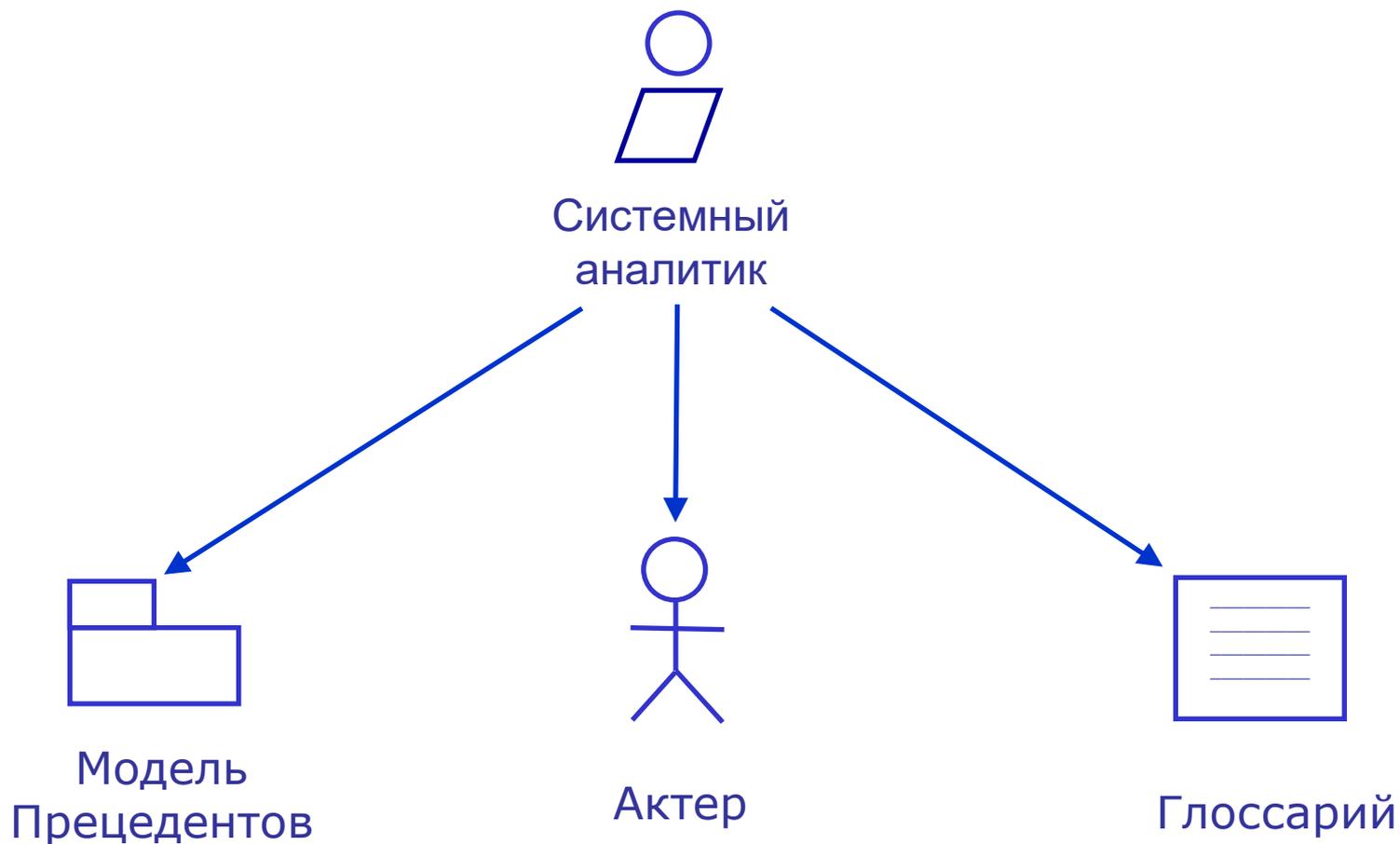
- Отвечает за все множество требований, которые моделируются как прецеденты, включая:
 - функциональные требования и
 - нефункциональные требования относящиеся к прецеденту
- Отвечает за ограничение системы
- Отвечает за поиск актеров и прецедентов
- Обеспечивает полноту и согласованность модели прецедентов
- Не отвечает за каждый конкретный прецедент (для этого есть **спецификатор прецедента**)

Построение модели прецедентов

Исполнители и артефакты (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Исполнители в потоке работ по построению модели прецедентов. Системный аналитик

Системный аналитик

- Отвечает за все множество требований, которые моделируются как прецеденты, включая:
 - функциональные требования и
 - нефункциональные требования относящиеся к прецеденту
- Отвечает за ограничение системы
- Отвечает за поиск актеров и прецедентов
- Обеспечивает полноту и согласованность модели прецедентов
- Не отвечает за каждый конкретный прецедент (для этого есть **спецификатор прецедента**)

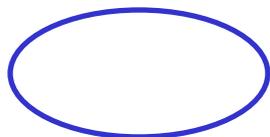
Исполнители в потоке работ по построению модели прецедентов. Спецификатор прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Спецификатор
прецедентов



Прецедент

Спецификатор прецедентов

- Отвечает за детальное описание одного и более прецедентов
- Спецификатор прецедентов должен работать с реальными пользователями и заказчиками системы
- Детально описывает на языке UML взаимодействие актеров и прецедентов

Исполнители в потоке работ по построению модели прецедентов. Проектировщик интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Проектировщик
интерфейса
пользователя



Прототип
интерфейса
пользователя

Проектировщик интерфейса пользователя

- Обычно разрабатывает по одному прототипу интерфейса пользователя для каждого актера

Исполнители в потоке работ по построению модели прецедентов. Архитектор

МГУ им. М.В.Ломоносова. Факультет ВМК.

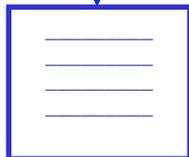
Романов Владимир Юрьевич ©2025



Архитектор

Архитектор

- Разрабатывает описание архитектуры в виде модели прецедентов



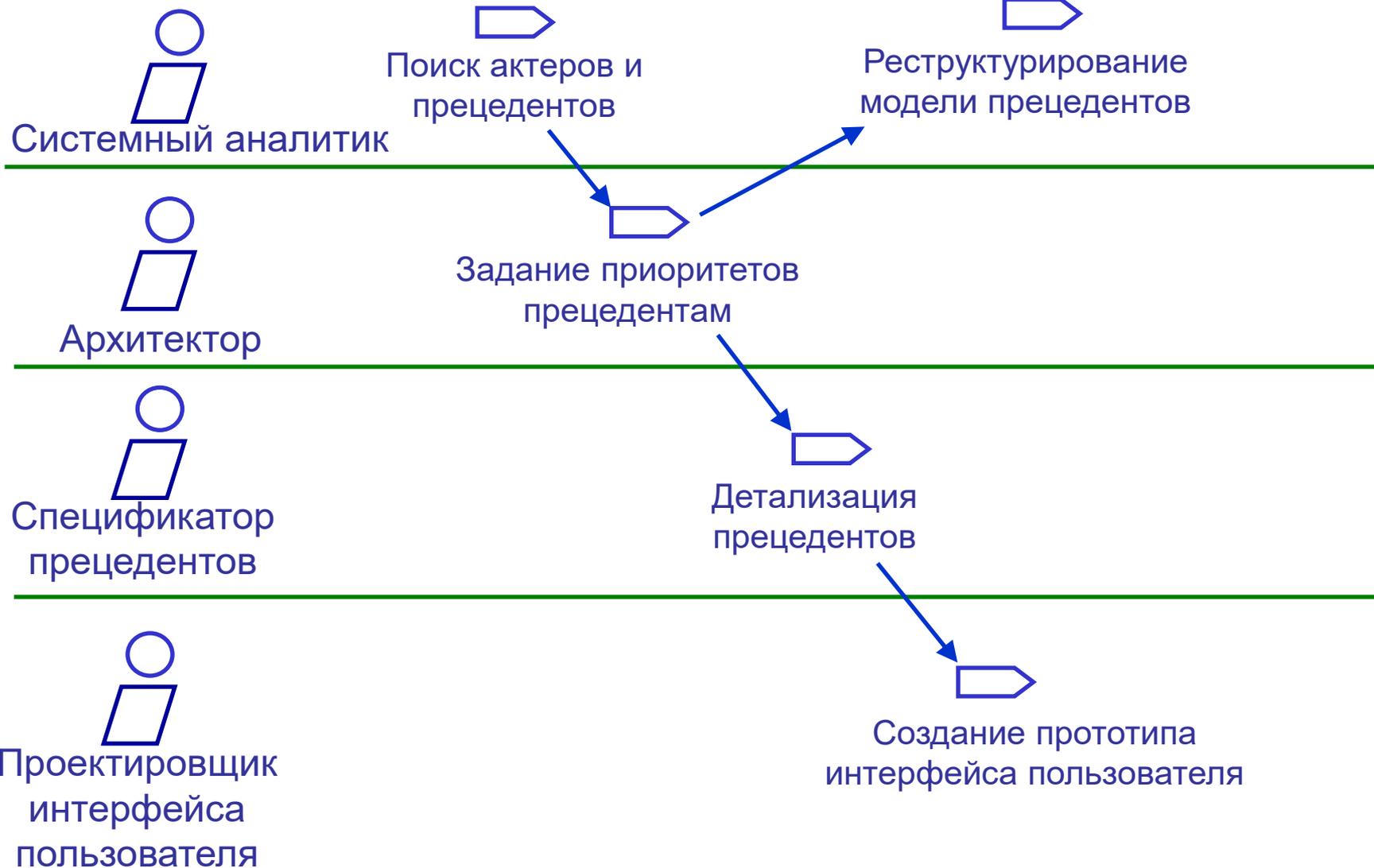
Описание
архитектуры

Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Обзор потока работ по построению модели прецедентов (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Каждая деятельность может быть пройдена несколько раз. Каждый раз при этом будет выполнена лишь часть работы текущей итерации.

1. **Системный аналитик** выполняет поиск актеров и прецедентов для подготовки новой версии модели прецедентов. На вход аналитику поступает:
 - Бизнес модель и модель предметной области (domain model)
 - Список возможностей (feature-list)
 - Дополнительные возможности (не функциональные)
2. **Архитектор** после этого задает приоритеты прецедентам, определяя прецеденты важные для создания архитектуры

Обзор потока работ по построению модели прецедентов (2)

3. Проектировщики интерфейса затем создают прототип интерфейса пользователя для каждого выявленного аналитиком актера на основе существующих прецедентов
4. Спецификаторы прецедентов выполняют детализацию прецедентов в соответствии с заданными архитектором приоритетами прецедентов
5. Системный аналитик выполняет реструктурирование модели прецедентов, определяя отношения обобщения и включения между прецедентами

Обзор потока работ по построению модели прецедентов (2)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Результат первого прохода через поток работ – 1-я версия артефактов:

- Модели прецедентов
- Прецеденты
- Прототипы пользователя

Следующие проходы через этот поток работ уточняют эти артефакты.

Различия в проходах по *потоку работ* на различных *фазах*

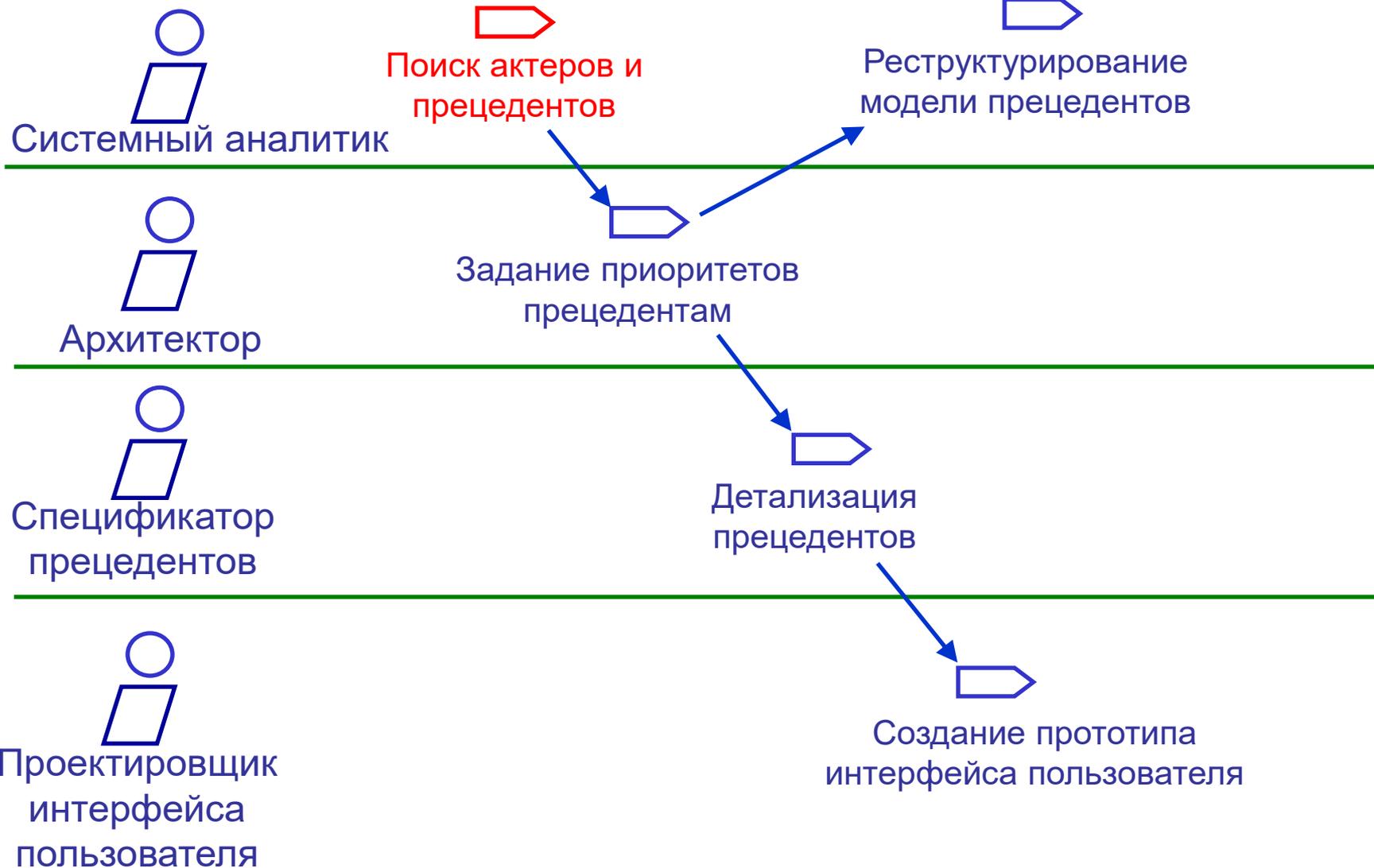
- В фазах **Inception** и **Elaboration** системный аналитик находит много актеров
- В фазе **Construction** изменения в составе актеров и прецедентов минимальны, но уточняются диаграммы прецедентов и взаимодействие прецедентов и актеров

Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Деятельность по поиску актеров и прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Цели:

1. Отделить систему от окружения
2. Кто или что взаимодействует с системой (актеры)
3. Какая функциональность ожидается от системы
4. Определить глоссарий понятий используемых при детальном описании прецедентов

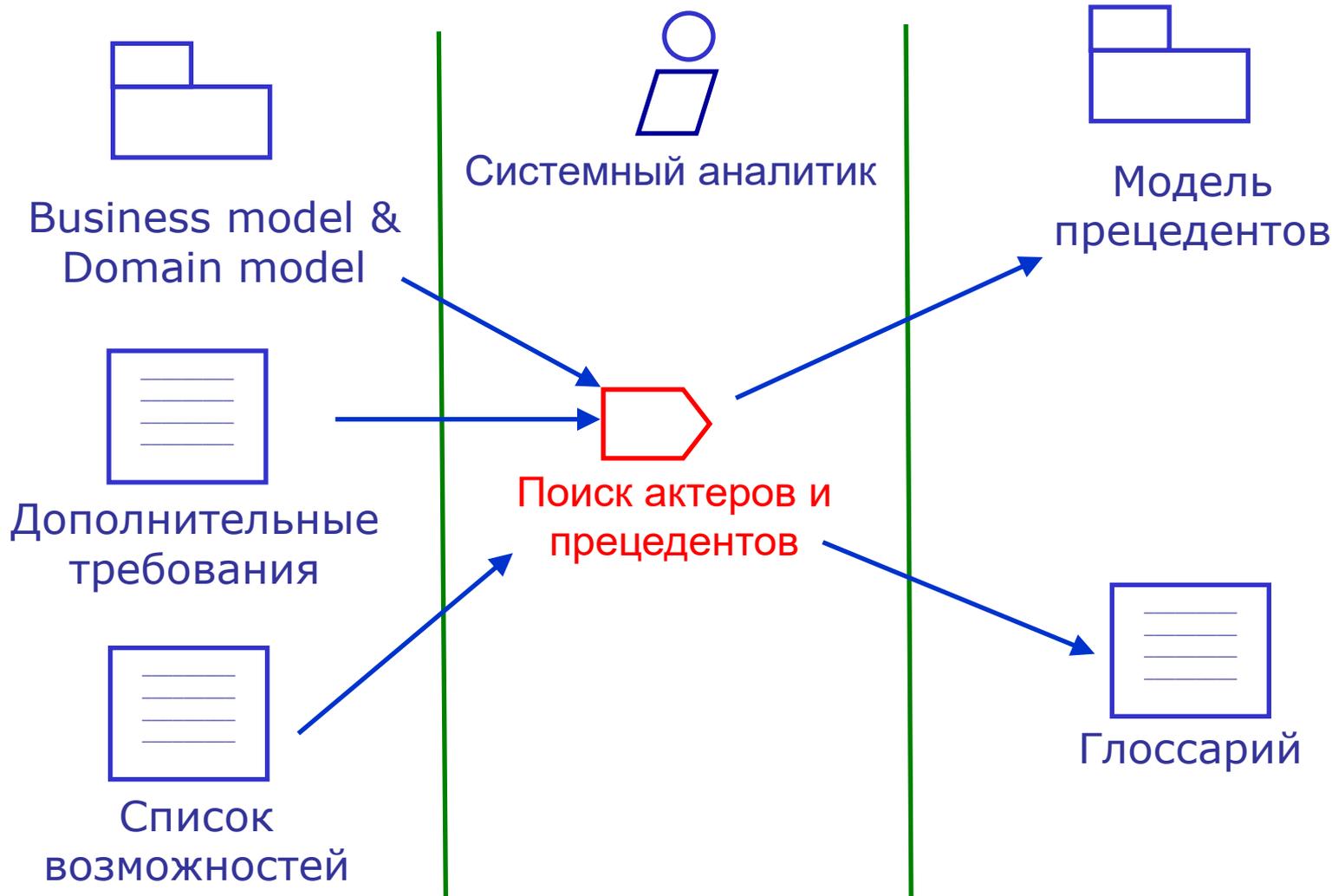
Поддеятельности:

1. Поиск актеров
2. Поиск прецедентов
3. Краткое описание каждого прецедента
4. Описание модели прецедентов в целом

Деятельность по поиску актеров и прецедентов и ее артефакты

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

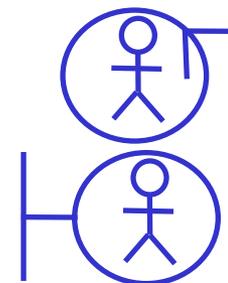


Поиск актеров

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. По одному актеру для каждого сотрудника (внутреннего актера) в бизнес-модели
2. По одному актеру для каждого бизнес актера (внешнего актера) в бизнес-модели



Критерии отбора:

1. Идентификация пользователей и их обобщение в актеров
2. Избегать пересечения ролей у актеров.
Если роли актеров пересекаются то ввести отношение обобщения у актеров

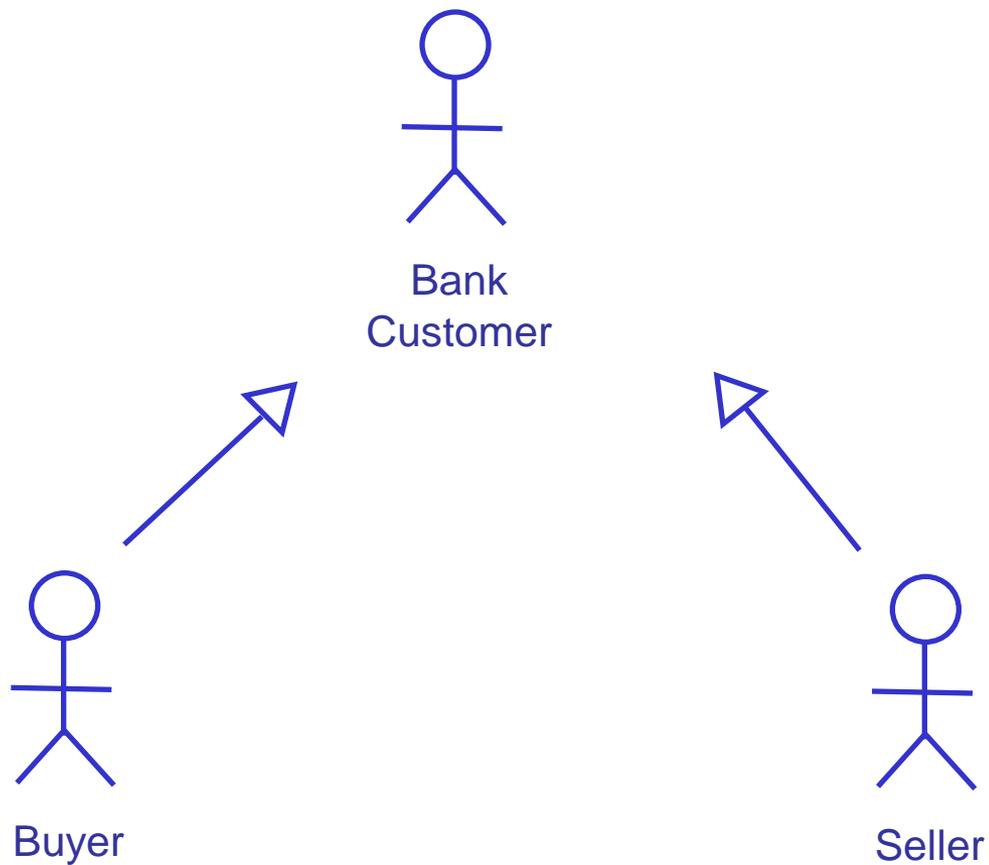
Результат поддеятельности:

1. Описание роли актера
2. Описание задач, для которых актер использует систему

Поиск актеров. Пример

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Поиск прецедентов

1. Прецедент предлагается для каждой роли каждого сотрудника участвующего в реализации бизнес-прецедента, если бизнес модель существует.
2. **Системный аналитик** предлагает для каждого актера *список прецедентов-кандидатов*. Каждый актер:
 - Создает, изменяет, отслеживает, удаляет бизнес-объекты. Например, заказы и счета.
 - Информировывает системы о событиях. Например, если актер – это таймер или календарь, а событие – счет просрочен.
 - Информировывает об истечении интервала времени
 - Актеры для:
 - Запуска системы
 - Остановки системы
 - Техобслуживание

Краткое описание каждого прецедента

1. Текстовое описание назначения каждого прецедента
2. Пошаговое описание деятельностей прецедента

Пример

1. Прецедент «оплатить счет» используется актером «покупатель» для пометки счетов к оплате.
2. Затем прецедент «оплатить счет» выполняет платеж в назначенный день

Описание модели прецедентов в целом

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Связь прецедентов друг с другом и с актерами.

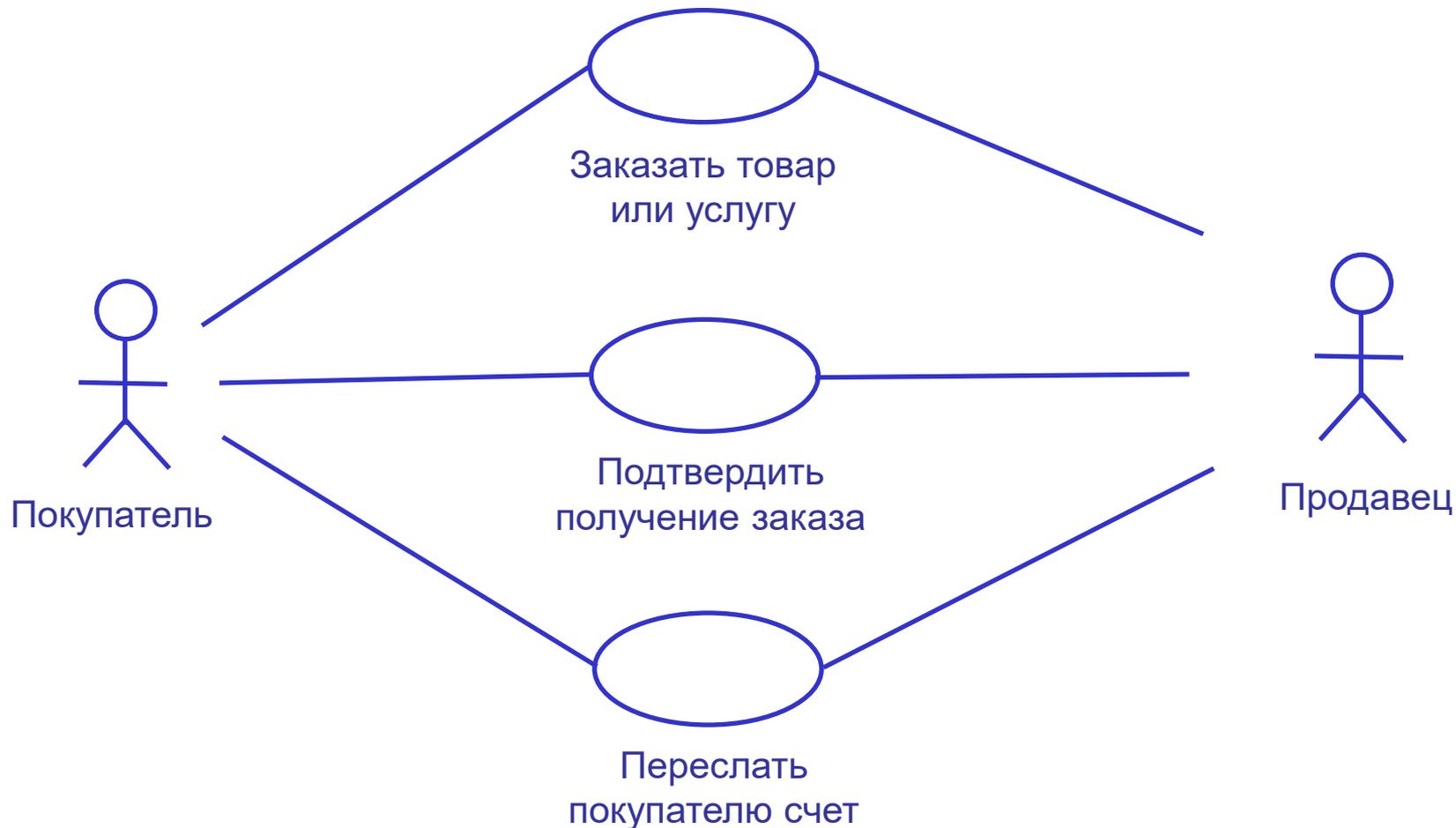
Диаграммы прецедентов представляющих модель прецедентов

- Диаграмму прецедентов на каждый бизнес-прецедент с описанием его реализации
- Прецеденты для каждого актера
- Группирование прецедентов в пакеты
 - Диаграмма прецедентов пакета
 - Диаграмма связей пакета

Пример диаграммы прецедентов. Отношения ассоциации

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Пример диаграммы прецедентов.

Отношения обобщения и расширения

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

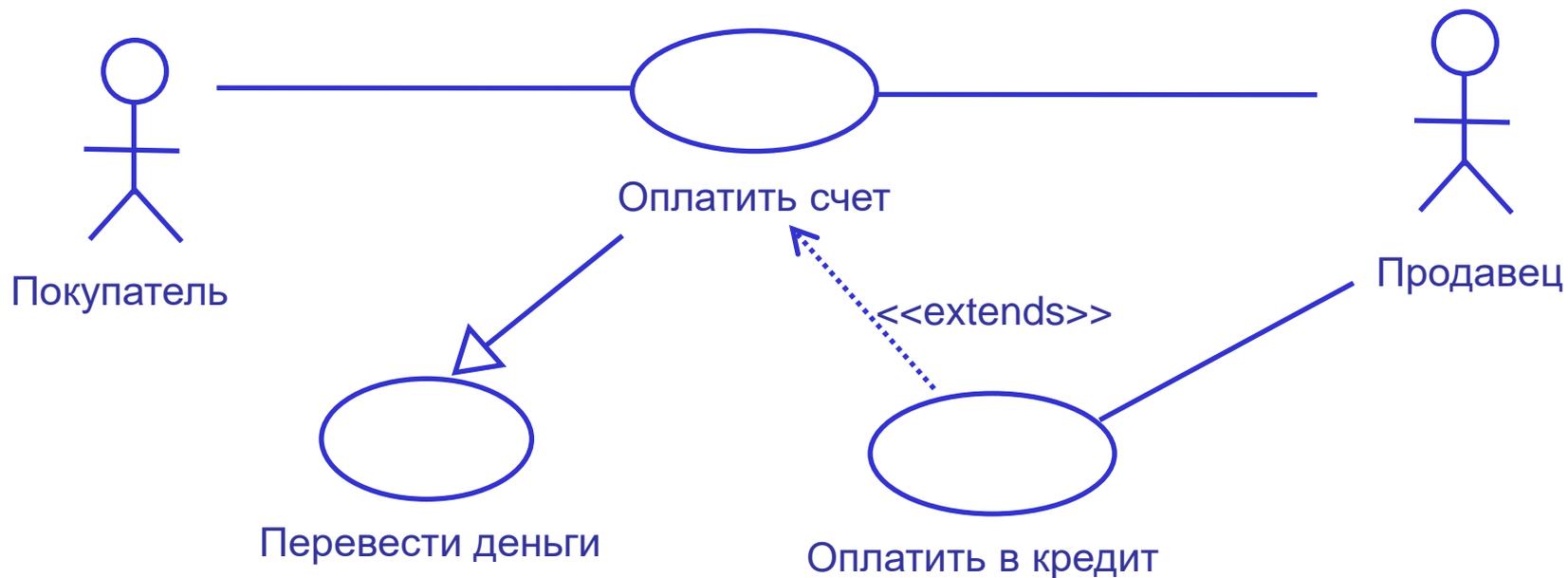
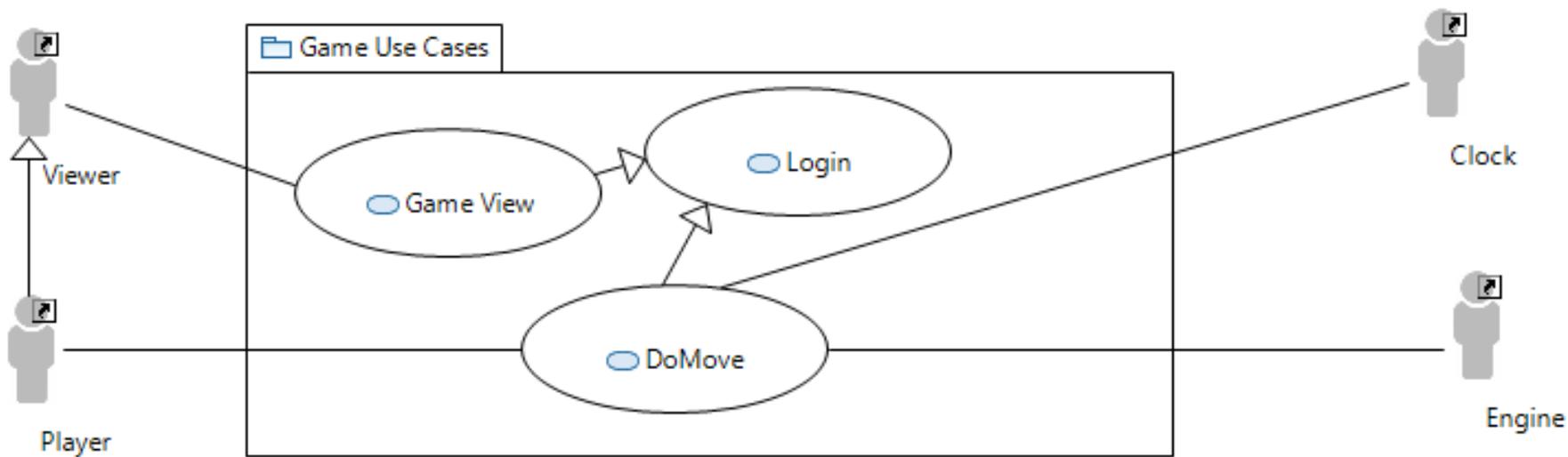


Диаграмма прецедентов. Настольные игры

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

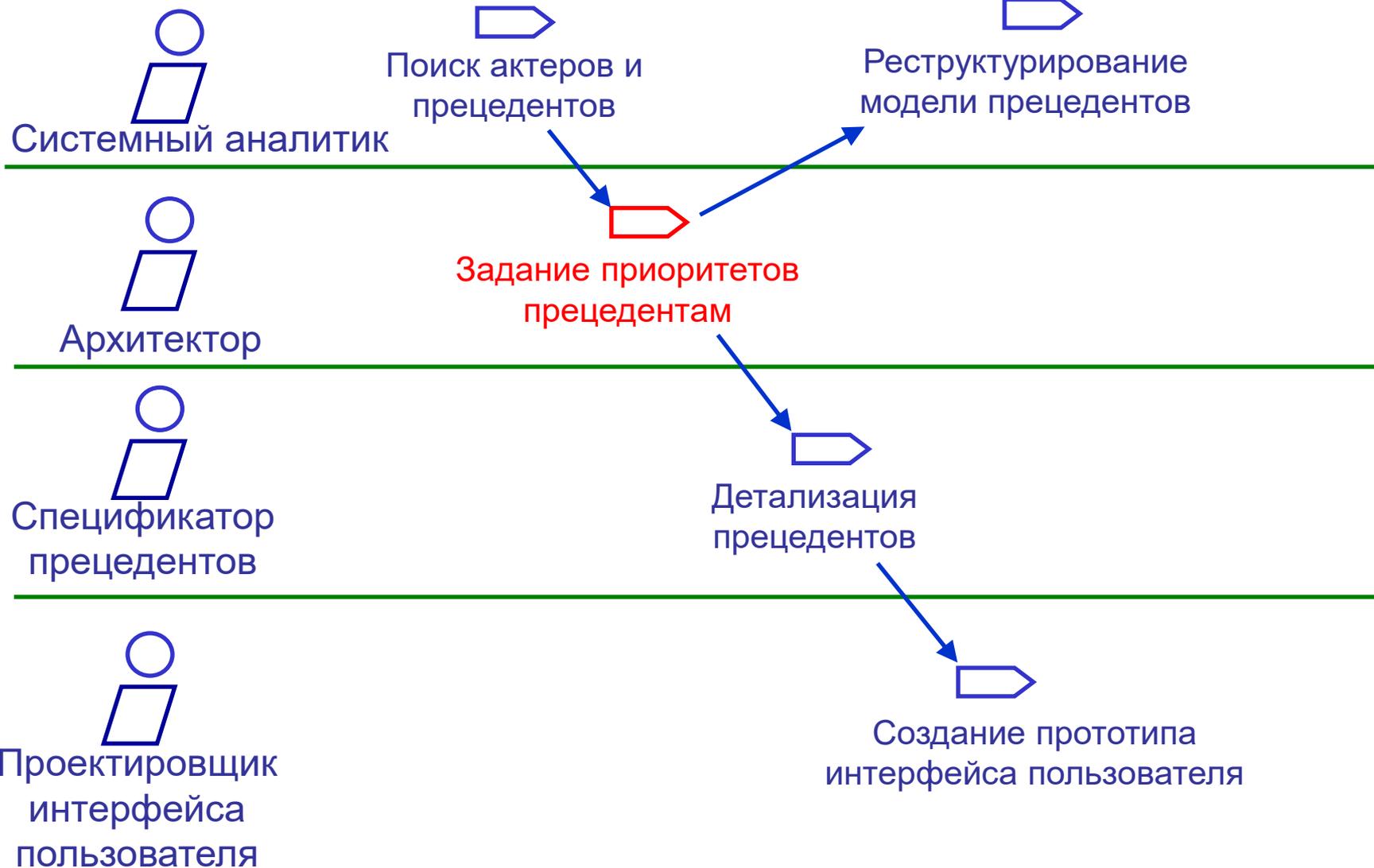


Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

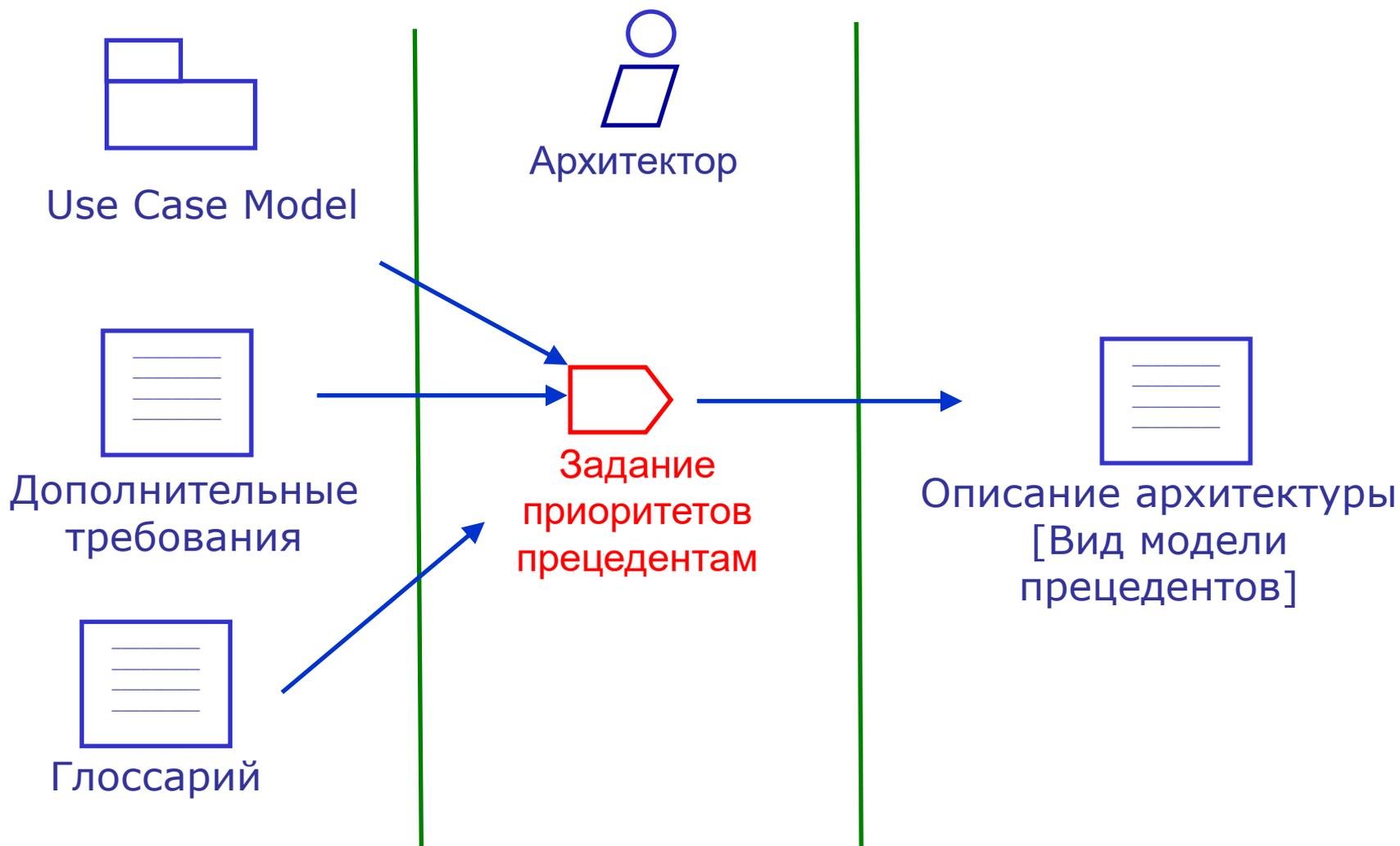
Романов Владимир Юрьевич ©2025



Деятельность по заданию приоритетов прецедентам

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Деятельность по заданию приоритетов прецедентам (1)

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

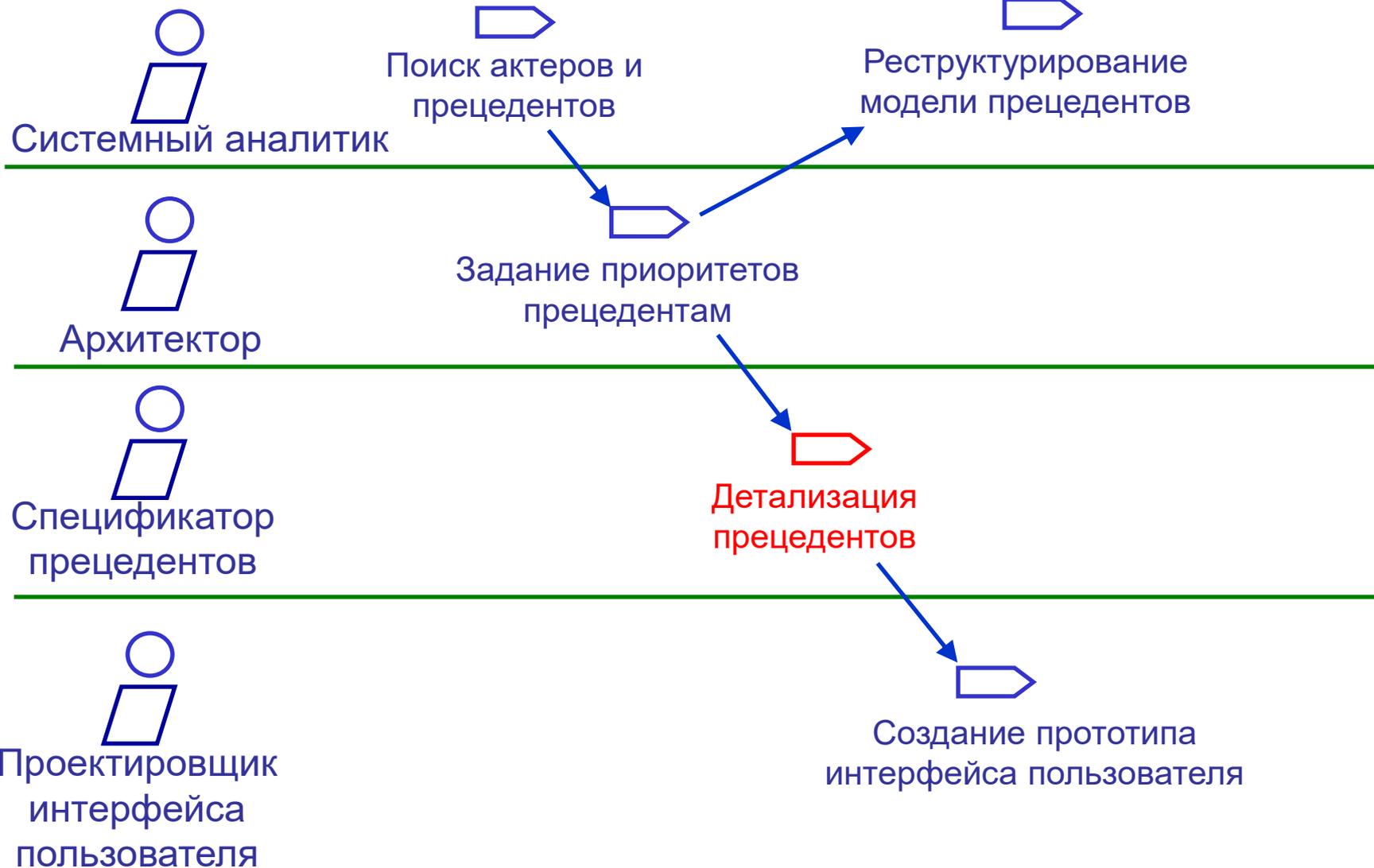
- 1. На начальных фазах** – определение прецедентов необходимых для раннего построения архитектуры.
- 2. На поздних фазах** - выбор прецедентов определяется очередностью и важностью прецедентов. Выбирают архитектор и менеджер проектов
- 3.** Результат фиксируется в описании архитектуры и используется менеджером проекта для планирования итераций

Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

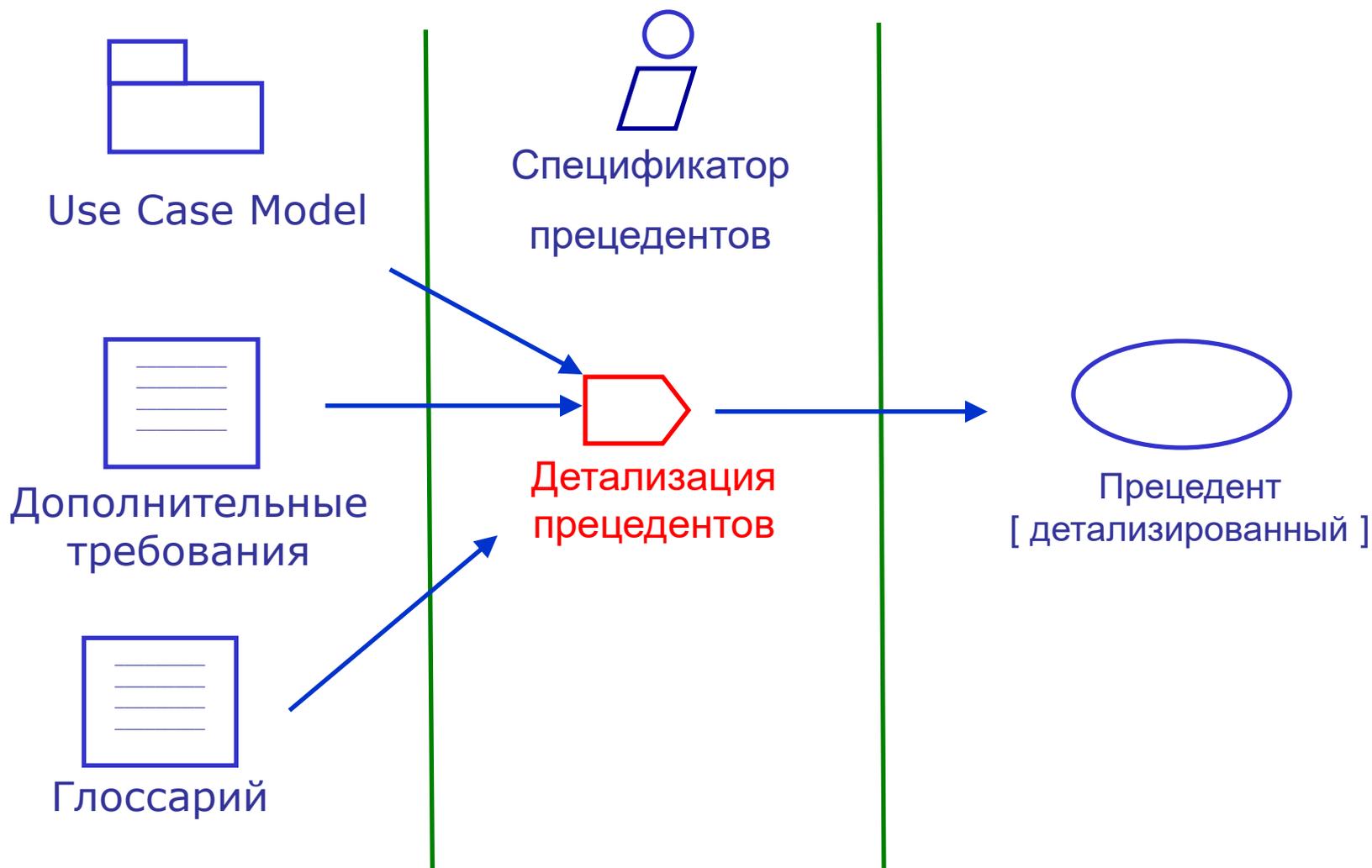
Романов Владимир Юрьевич ©2025



Деятельность по заданию детализации прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Деятельность по детализации прецедентов

1. Пошагово описывается последовательность действий
2. Детализация описания прецедента
3. Формализация описания прецедента

Спецификатор прецедентов непосредственно работает с будущими пользователями прецедента.
Должна быть обратная связь от пользователя.

Деятельность по детализации прецедентов. Структурирование описание прецедента

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

Определяются **состояния** экземпляров прецедента
и переходов между ними

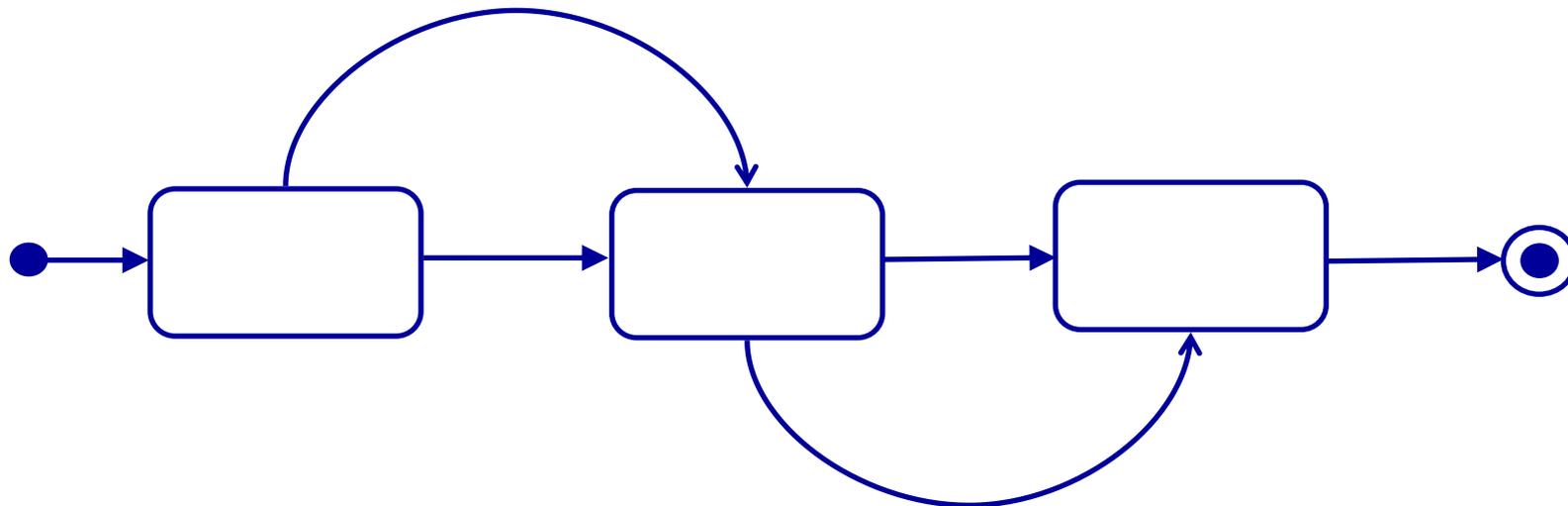
1. Выбирается **полный основной путь** от начального состояния до конечного состояния
2. Описываются **альтернативные пути**
 - Отказы актера
 - Вмешательство других актеров
 - Ошибочный ввод от актеров
 - Недостаток ресурсов

Деятельность по детализации прецедентов

Пример диаграммы

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Переход – последовательность действий, которая выполняется экземпляром прецедента в случае некоторого события

Состояния – например, состояния документа

Деятельность по детализации прецедентов. Формализованное описание прецедента

МГУ им. М.В.Ломоносова. Факультет ВМК.

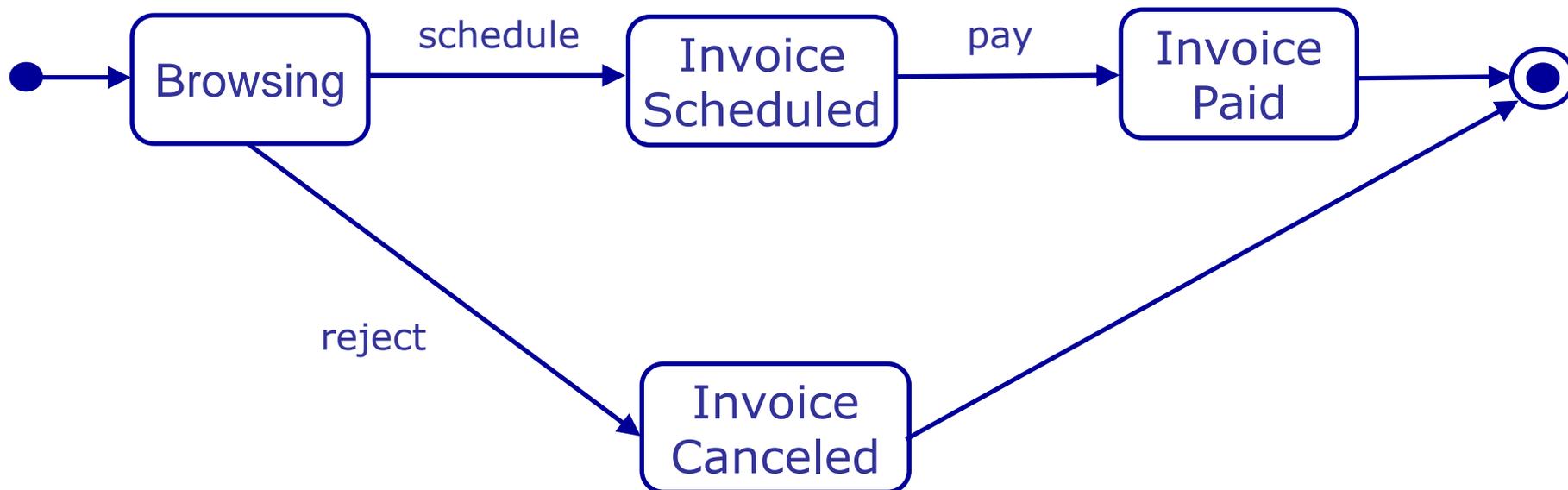
Романов Владимир Юрьевич ©2025

1. **State transition diagram** (*диаграмма переходов и состояний*) – для описания состояния прецедента
2. **Activity diagram** (*диаграмма деятельности*) – для детального описания переходов как последовательности действий
3. **Collaboration diagram & Sequence diagram** (*диаграмма кооперации объектов и диаграмма последовательности взаимодействия объектов*) – для описания взаимодействия экземпляра актера и экземпляра прецедента

Деятельность по детализации прецедентов. Пример формализованного описания

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

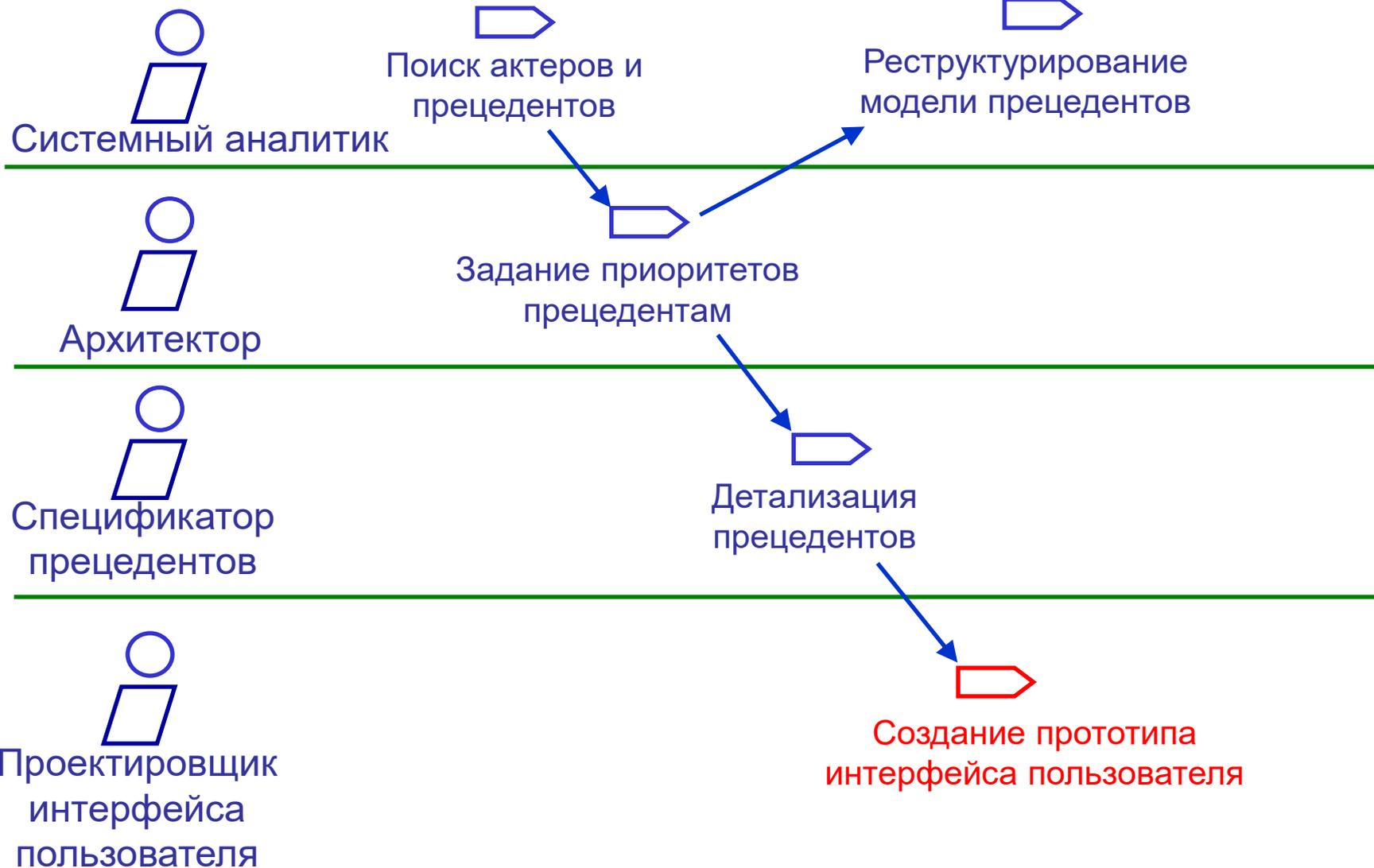


Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

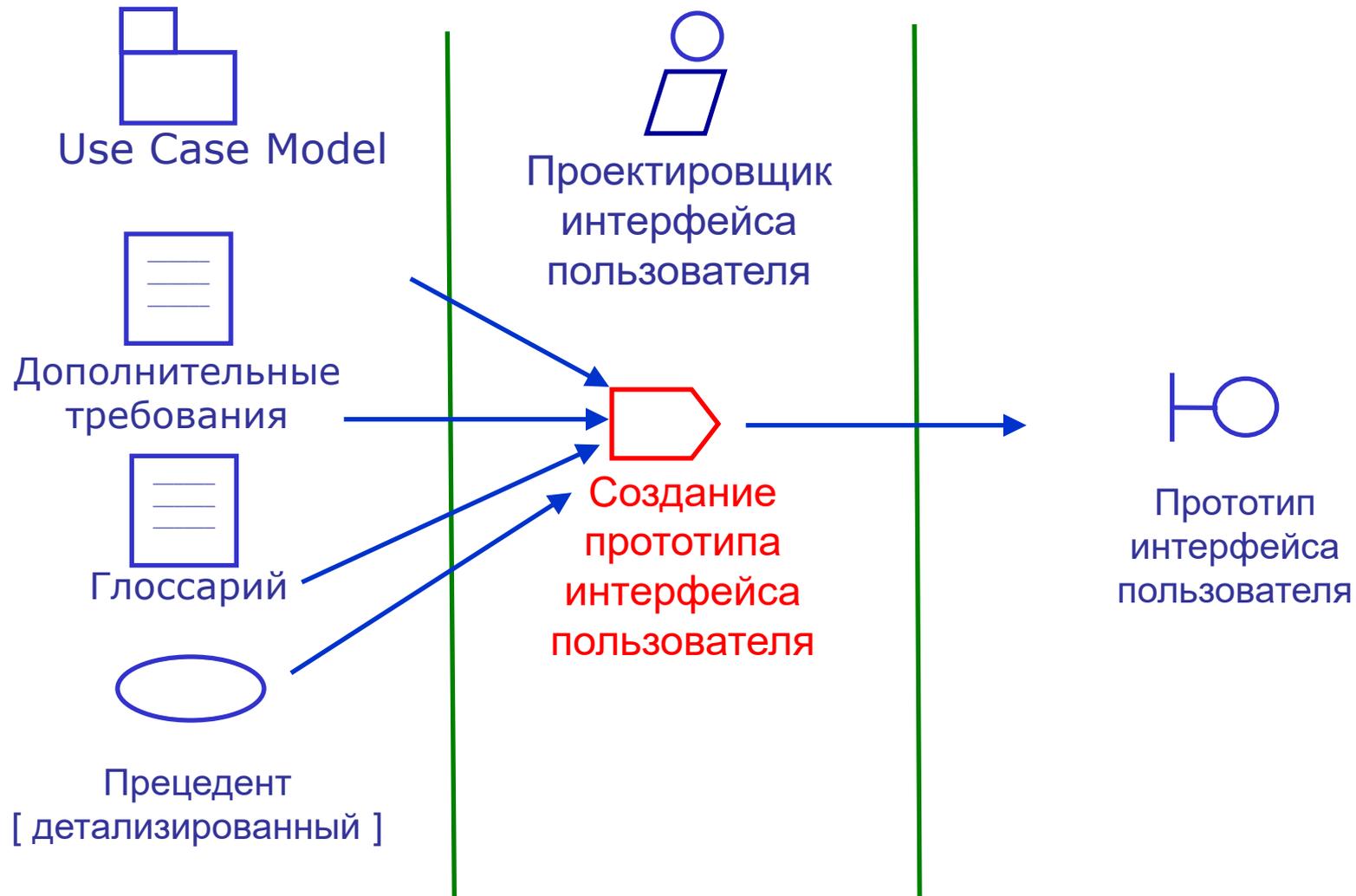
Романов Владимир Юрьевич ©2025



Создание прототипа интерфейса пользователя

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Создание прототипа интерфейса пользователя.

Создание логического экрана

- Определяет что необходимо от интерфейса пользователя, что бы сделать доступным прецедент для актера.
- Элементы интерфейса пользователя – это атрибуты прецедента.
- Актер манипулирует атрибутами прецедента.
- Часто атрибуты – это термины из глоссария.
- Атрибуты представляются пиктограммами, списками, папками, объектами на карте

Создание прототипа интерфейса пользователя. Создание физического экрана

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

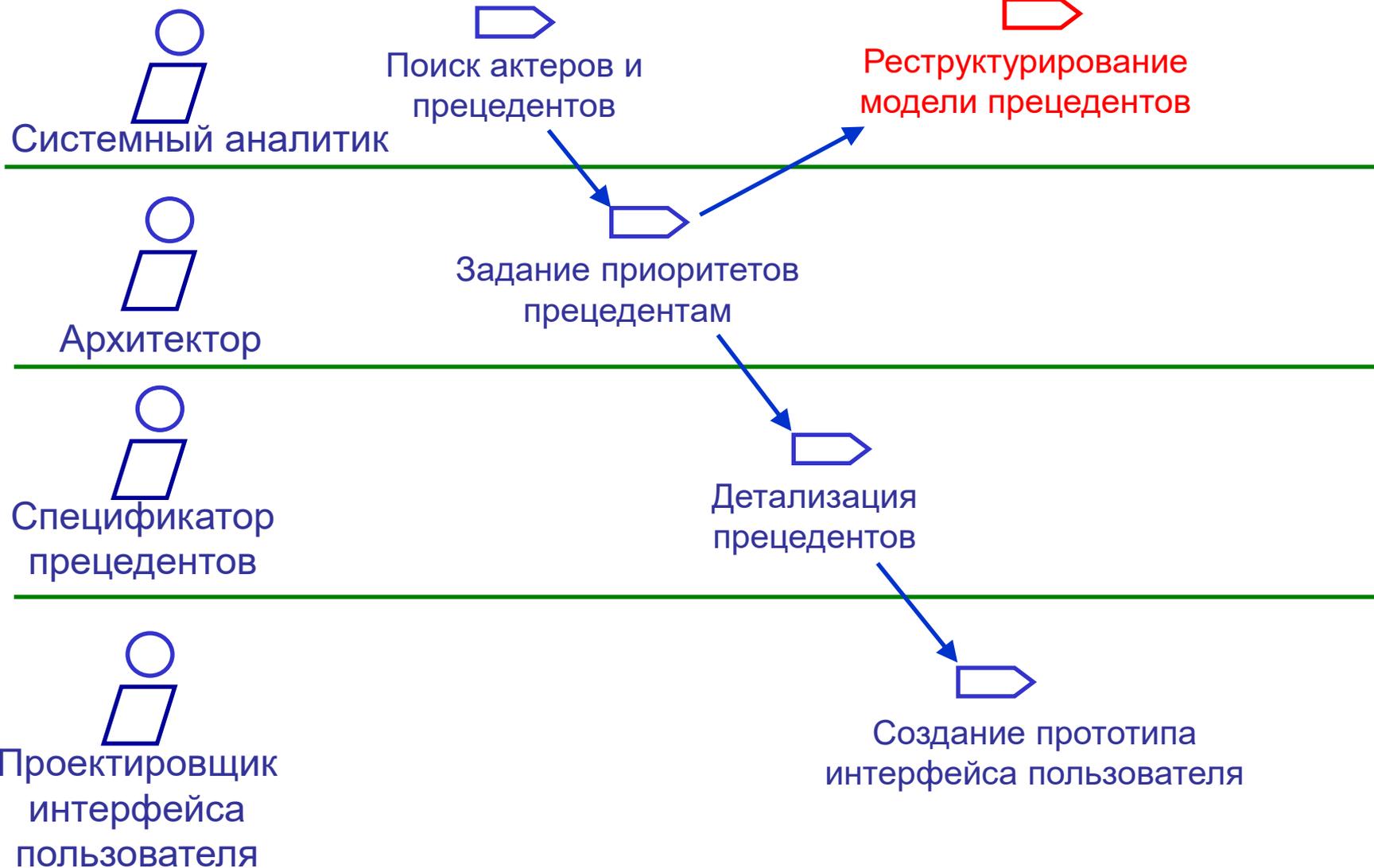
- Взаимное расположение управляющих элементов
- Для навигации и структурирования – папки, окна, инструменты (tools)
- Табуляция, ускорители
- Цвета, размеры

Построение модели прецедентов

Поток работ (workflow)

МГУ им. М.В.Ломоносова. Факультет ВМК.

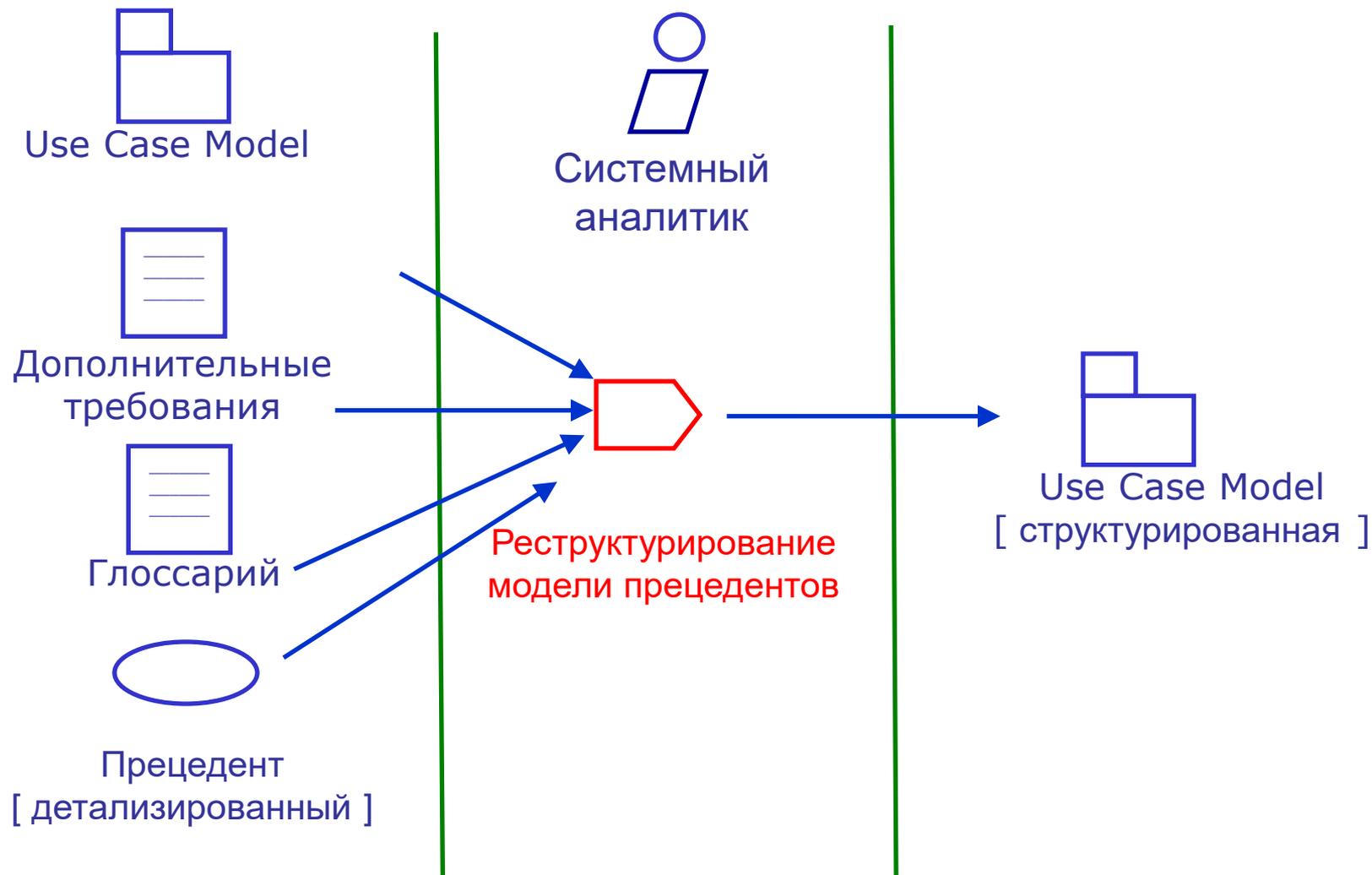
Романов Владимир Юрьевич ©2025



Реструктурирование модели прецедентов

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Реструктурирование модели прецедентов. Разделяемые описания функциональности.

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Выделить **общие** и **совместно используемые** прецеденты (описания функциональности которые могут быть использованы более специализированно)
2. Экземпляры в обобщенных прецедентах могут выполнить **все** поведение, описанное в обобщающем прецеденте

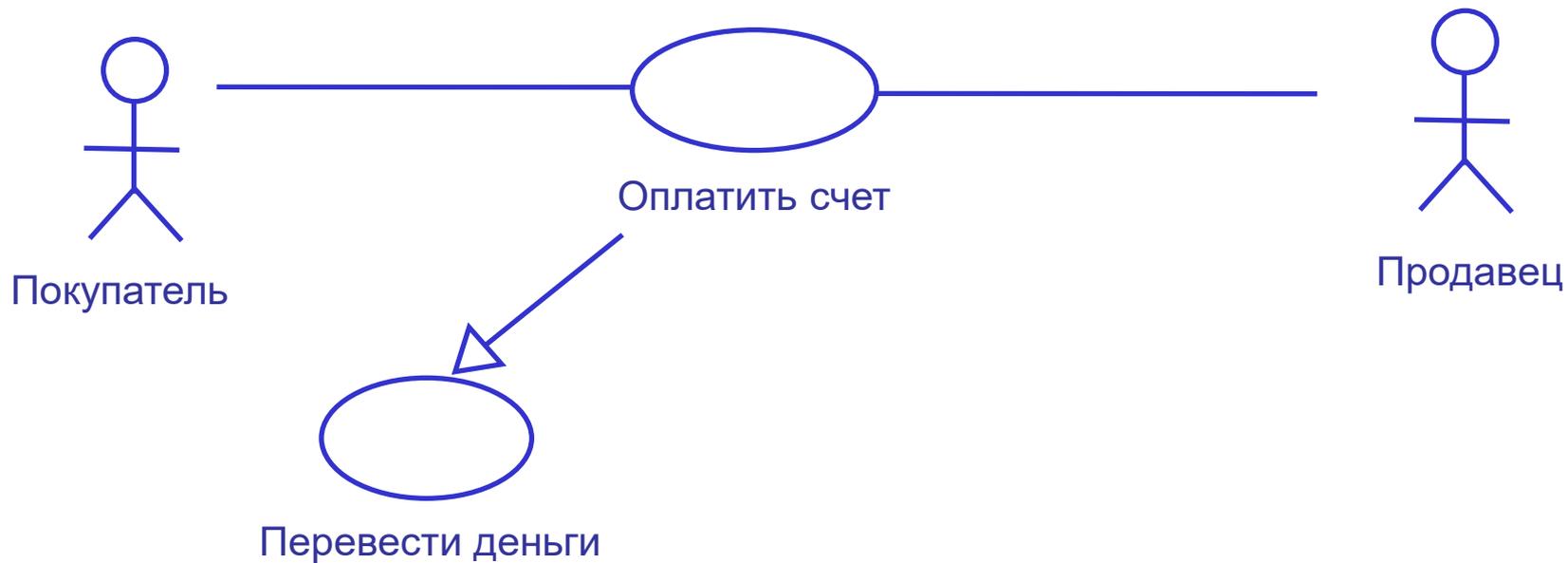


3. Экземпляр **A** будет включать поведение **B**

Реструктурирование модели прецедентов. Разделяемые описания функциональности. Пример

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025



Реструктурирование модели прецедентов.

Дополнительные и условные описания функциональности

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

1. Выделить **дополнительные** и **условные** прецеденты которые могут расширить более специфичные прецеденты
2. Отношение зависимости со стереотипом <<extends>> моделируют дополнение к последовательности действий прецедента. Отношение включает условие расширения и точку расширения в целевом прецеденте.



3. Экземпляр **A** может при некоторых условиях включать поведение **B**

Реструктурирование модели прецедентов.

Дополнительные и условные описания функциональности

МГУ им. М.В.Ломоносова. Факультет ВМК.

Романов Владимир Юрьевич ©2025

