**M2**

Вариант 5

1. Какие отношения разрешены между элементами диаграммы вариантов использования?
2. В чем отличие документа ТЗ от документа SRS?
3. Расставьте отношения на диаграмме



1. Зачем определять приоритеты требований?
2. Расположите в хронологическом порядке работы, выполняемые в процессе внесения изменений в спецификацию требований:

    А) Внесение изменений в код системы

    Б) Разработка требования

    В) Анализ требования, его стоимость

    Г) Внесение изменения в спецификацию

    Д) Оформление изменения

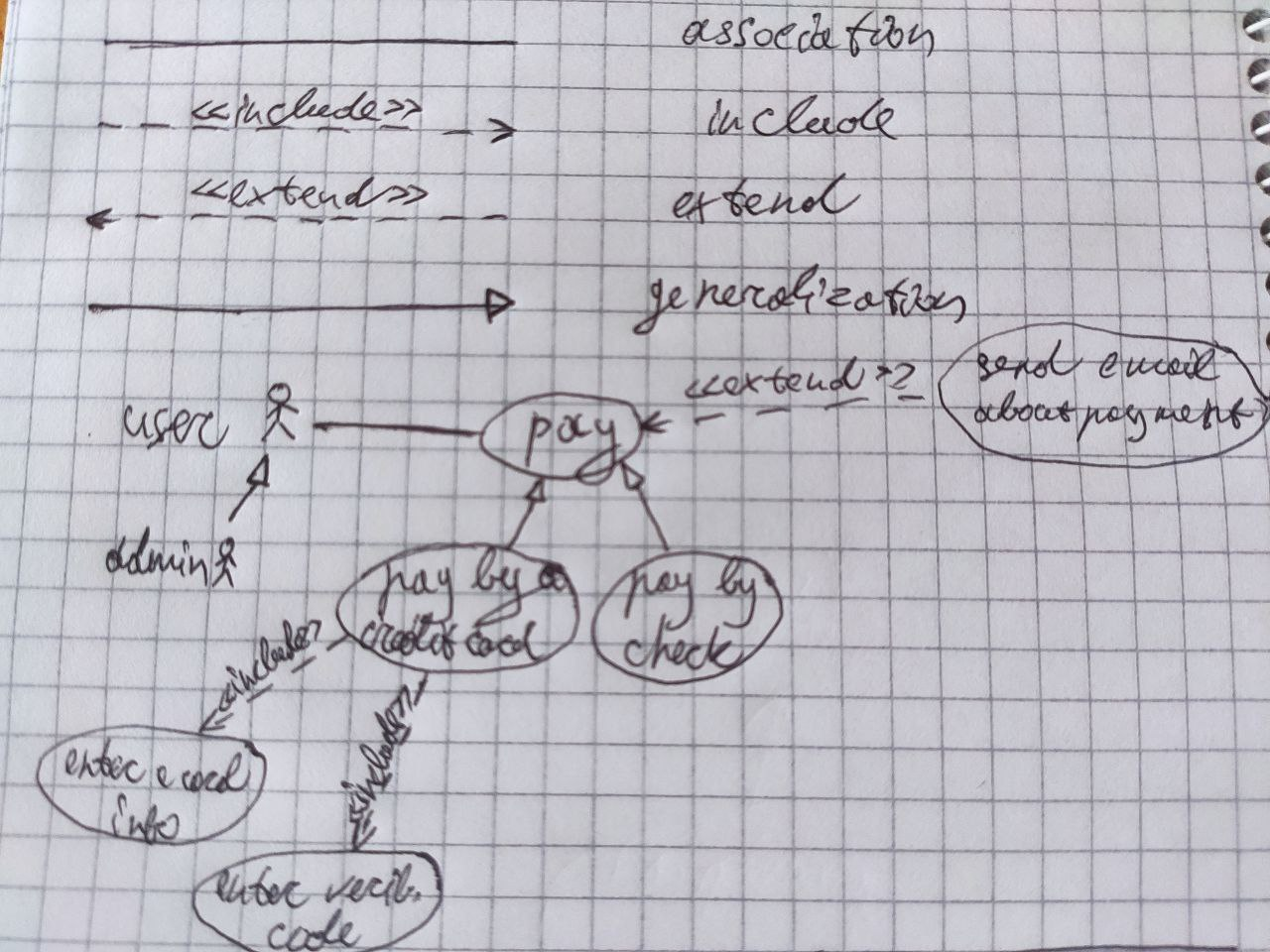
1. Перечислите проблемы, возникающие при пропуске классов пользователей.
2. Обобщение возможно между:
   1. актером - вариантом использования
   2. вариантом использования - вариантом использования
   3. актер-актер
   4. все варианты правильные
3. Перечислите цели прототипирования
4. Поясните понятие одноразового прототипа и цели его создания.

|  |  |  |
| --- | --- | --- |
|  | Включение | Расширение |
| Является ли прецедент обязательным? | Yes | No |
| Является базовый прецедент полным без данного прецедента? | No | Yes |

1. По используемым средствам проверку требований классифицируют:
   1. тесты
   2. критерий приемлемости
   3. экспертиза
   4. просмотр
   5. прототип
   6. сквозной контроль

1. Какие отношения разрешены между элементами диаграммы вариантов использования?

1. include (включение)
2. exclude (расширение)
3. Generalization (обобщение)
4. Association

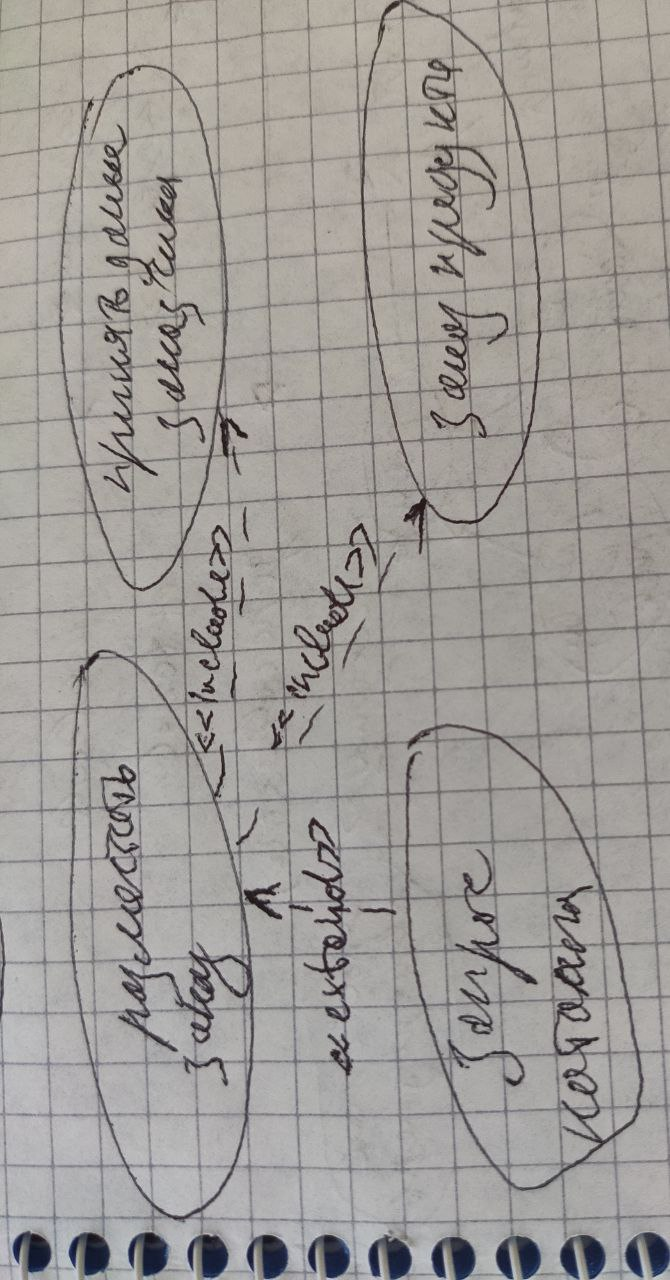


2.

SRS - структурированный набор требований (функциональность, производительность, конструктивные ограничения и атрибуты) к программному обеспечению и его внешним интерфейсам (IEEE 980)

ТЗ (technical requirements document, TRD) содержит требования к программному, аппаратному обеспечению, требованиям к платформе, языки программирования, на которые ПО будет реализовано. Может содержать разделы: краткое описание проекта, риски, функциональные и нефункциональные требования.

3.



4. Для того, чтобы правильно распределить расходы на разработку различных частей системы. Если этого не делать может получиться, что разработчики много времени будут делать то, что в общем и не нужно ни заказчику, ни пользователям, в то время как сделать плохо либо вообще не сделать то что важно.

5. Расположите в хронологическом порядке работы, выполняемые в процессе внесения изменений в спецификацию требований:

Д) Оформление изменения

В) Анализ требования, его стоимость

Б) Разработка требования

Г) Внесение изменения в спецификацию

А) Внесение изменений в код системы

6. Перечислите проблемы, возникающие при пропуске классов пользователей

Вигерс рекомендует выделить максимальное количество классов пользователей, даже если их окажется несколько десятков. А потом сократить максимум до 15 классов (объединяя похожие классы в один, либо рассматривая некоторые классы как подклассы других).

Но если при этом не учесть какую-то группу пользователей, то и разрабатываться система будет без их учета. Следовательно, в худшем случае эти пользователи просто не смогут использовать разрабатываемое ПО – то есть, потеря пользователей, а следовательно, и денег.

7. Обобщение возможно между:

b. вариантом использования - вариантом использования

c. актер-актер

8. Перечислите цели прототипирования

- убедиться, что все участники разработки правильно поняли заказчика

- благодаря чему также уменьшается стоимость разработки (точнее не увеличивается, из-за отсеивания несоответствий на начальных этапах)

- позволяет пользователям увидеть, как будет работать программа

Но и риски создания прототипа идут неразрывно с целями и выгодами:

- риск попадания плохо реализованных фрагментов в реальный процесс разработки

- трата время на его создания

- недостаточный анализ

- также возможно непонимание заказчика (почему продукт должен делаться так долго, когда за пару дней сделали уже работающую програму)

9. Поясните понятие одноразового прототипа и цели его создания.

Прототип главным параметром целью которого является скорость. При создании одноразового прототипа создается прототип, показывающий определенные компоненты системы, при этом он должен создавать быстро. Он даже может создаваться, используя отличные технологии и языки от того, что будут использоваться при реализации.

Главное правило – он должен быть отброшен. Это важно с обоих сторон:

* + 1. Прототип во время реализации воспринимается как прототип, поэтому нельзя быть уверенным в адекватности многих принимаемых во время реализации решений
    2. Понимание того, что прототип будет отброшен позволяет увеличить скорость максимально, так как разработчик не будет думать о необходимости дальнейшем развитии прототипа

10. По используемым средствам проверку требований классифицируют:

a. тесты

b. критерий приемлемости

e. прототип