

Разработка новой МИС

... захват рынка и
сопровождение

Сысоев С.С.
ООО "ПетроМС"



Защита на 4-й уровень ТРИЗ, 2025

Цели и задачи проекта



Цель

Создание нового доходного направления в бизнесе с нуля

Задачи

Определение рыночной ниши

Формирование команды

Сбор аналитики

Разработка

Поиск клиентов

Ограничения

Сжатые сроки / ограниченный бюджет



1. Знакомство с ТРИЗ
2. Команда проекта
3. Преподавание - доступ к кадрам
4. Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам
5. Готовая аналитика по некоторым общим частям МИС
6. Накопления от предыдущей деятельности

Использование ресурсов



1. Знакомство с ТРИЗ
2. Команда проекта
3. Преподавание - доступ к кадрам
4. **Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам**
5. **Готовая аналитика по некоторым общим частям МИС**
6. Накопления от предыдущей деятельности

Задачи

Определение рыночной ниши

Формирование команды

Сбор аналитики

Разработка

Поиск клиентов

Использование ресурсов



1. Знакомство с ТРИЗ
2. **Команда проекта**
3. **Преподавание** - доступ к кадрам
4. Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам
5. Готовая аналитика по некоторым общим частям МИС
6. **Накопления от предыдущей деятельности**

Задачи

Определение рыночной ниши

Формирование команды

Сбор аналитики

Разработка

Поиск клиентов

Использование ресурсов



1. Знакомство с ТРИЗ
2. Команда проекта
3. Преподавание - доступ к кадрам
4. **Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам**
5. **Готовая аналитика по некоторым общим частям МИС**
6. Накопления от предыдущей деятельности

Задачи

Определение рыночной ниши

Формирование команды

Сбор аналитики

Разработка

Поиск клиентов

Использование ресурсов



1. **Знакомство с ТРИЗ**
2. **Команда проекта**
3. **Преподавание** - доступ к кадрам
4. Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам
5. **Готовая аналитика по некоторым общим частям МИС**
6. **Накопления от предыдущей деятельности**

Задачи

Определение рыночной ниши

Формирование команды

Сбор аналитики

Разработка

Поиск клиентов

Использование ресурсов



1. Знакомство с ТРИЗ
2. Команда проекта
3. Преподавание - доступ к кадрам
4. **Сопровождение предыдущего (устаревшего) ПО - доступ к потенциальным Заказчикам**
5. Готовая аналитика по некоторым общим частям МИС
6. Накопления от предыдущей деятельности

Задачи

Определение рыночной ниши
Формирование команды
Сбор аналитики
Разработка

Поиск клиентов

Команда проекта

Велюхов Ю.Г.
Ермаков А.В.
Куприянов А.С.
Афиногенова Е.В.
Бакеева К.В.
Сысоев С.С.

Бакеев Р.С.
Моисеев Ю.А.
Афиногенов В.А.
Скворцов А.В.
Колабутин Н.В.
Шамшурина Е.А.
Григоренко П.В.
Сысоев С.С. - младший

Сотрудники Заказчиков
Студенты-дипломники



Результаты

2016 г. 2 внедрения
2019 г. 10 внедрений
2024 г. 17 внедрений

Объем рынка СПб: 27 медицинских учреждений



Интересные задачи



Управление вирусным маркетингом

ЕСЛИ стимулировать Заказчиков обсуждать систему с коллегами,
ТО увеличивается степень информированности рынка о продукте,
НО падает доверие к информации

Приемы:

- 2. Вынесения
- 7. Матрешки
- 24. Посредника

Принципы:

Системный переход

ИКР:

Глав. врач САМ хочет продемонстрировать систему

Интересные задачи



Бюрократический гамбит

Генеральный подрядчик не выполнил свою часть работ, но отказывается **официально** признавать это.

Разработка субподрядчика стоит на месте. Сдача работ невозможна без интеграции с подсистемой, разрабатываемой генеральным подрядчиком.

Генеральный подрядчик избегает общения с субподрядчиками.

Приемы:

13. Наоборот

ВПР:

Официальный
документооборот

ИКР:

Генеральный подрядчик САМ ищет
контакт с субподрядчиком и берёт
решение проблемы на себя

Интересные задачи

Сопровождать нельзя забить

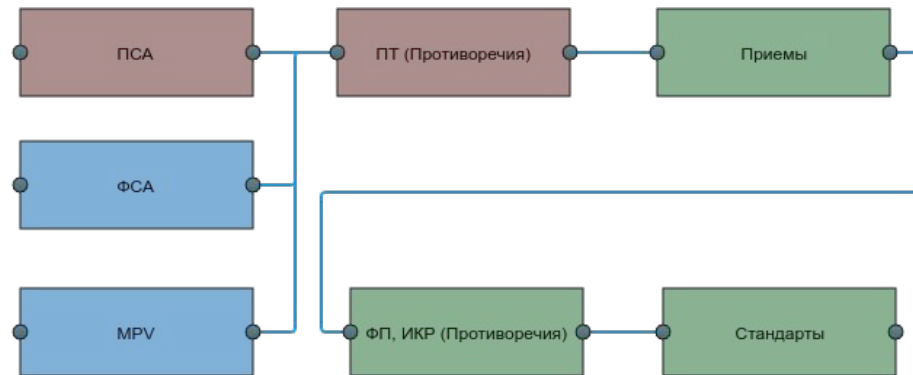
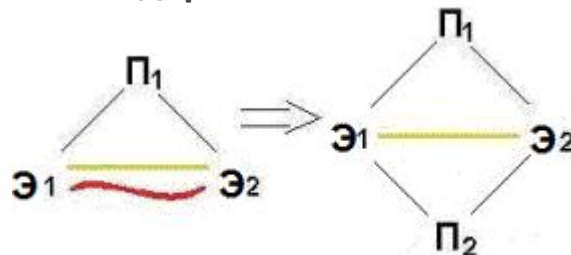
ЕСЛИ хорошо, быстро и качественно выполнять заявки на сопровождение,
ТО Заказчики довольны и охотно рекомендуют систему коллегам,
НО количество заявок становится больше, а процесс сопровождения дороже

Принципы:

Системный переход

В надсистеме заявок выполняется много, а в подсистеме - мало

Стандарты:



ИКР:

Заказчик САМ выполняет свои заявки

Шаг назад от ИКР:

Заказчик выполняет значительную часть заявки

Интересные задачи



Учебные задачи, созданные на основе реальных ситуаций

1. Вредная кнопка
2. Скрытный принтер
3. Неправильный граф
4. Передача закладных
5. Незадачливый кассир
6. Ленивые ортопеды
7. Петина библиотека
8. ...

Спасибо за внимание!



Сысоев Сергей Сергеевич
ООО "ПетроМС"
sysoev@petroms.ru





Вредная кнопка

В приложении на базе Windows Forms есть кнопка “Построить отчеты”. Нажатие на эту кнопку вызывает процедуру-обработчик, которая работает 5 минут. На время работы этой процедуры кнопка делается неактивной (чтобы пользователь не нажимал на нее несколько раз подряд).

Пользователь (по разным индивидуальным причинам) все равно желает ткнуть несколько раз мышью в неактивную кнопку. Поскольку приложение “висит” в обработчике, ОС Windows буферизует эти нажатия, и передает их приложению, когда обработчик завершается, и кнопка становится активна.

Эффект, наблюдаемый пользователем, резко расходится с его ожиданиями (многократный вызов обработчика может остановить работу на несколько часов).

Как исправить ситуацию?



Вредная кнопка - решение

Стандартный подход заключается в вынесении обработчика нажатия на кнопку в отдельный поток исполнения. В результате этого приложение во время работы обработчика не “повисает”, и нажатия на кнопку не буферизуются ОС, а сразу передаются приложению. Приложение, определив, что в данный момент еще не закончена обработка предыдущего нажатия, просто игнорирует новые нажатия.

Возможен более простой подход (мини-задача). Оперативная зона - пространство, занимаемое кнопкой, должно быть доступно пользователю (чтобы он мог нажать кнопку и построить отчет) и должно быть недоступно, когда отчет строится. Разделение противоречащих свойств во времени.

Кнопка выносится в пункт выпадающего меню. Когда приложение “висит”, пользователь не может открыть меню и нажать кнопку.



Скрытный принтер

В инфомате, выдающем номерки электронной очереди, установлен чековый принтер, печатающий выданные номерки.

Процедура выдачи номерка выглядит так:

1. Пользователь нажимает на экране инфомата кнопку “Выдать номерок”.
2. Обработчик кнопки проверяет, все ли хорошо с принтером.
3. Если принтер готов, новый номерок регистрируется в БД, после чего отправляется на печать.

Оказывается, если в принтере заканчивается лента, он не спешит сообщить об этом в ОС (в среднем принтер скрывает эту ситуацию около 30 секунд). За это время может быть зарегистрировано 2-3 “глухих” номерка.

Как быть?



Скрытный принтер - решение

Оперативное время = $B1 + B2$

$B1$ - 30 секунд, когда принтер уже не работает, но система об этом еще не знает.

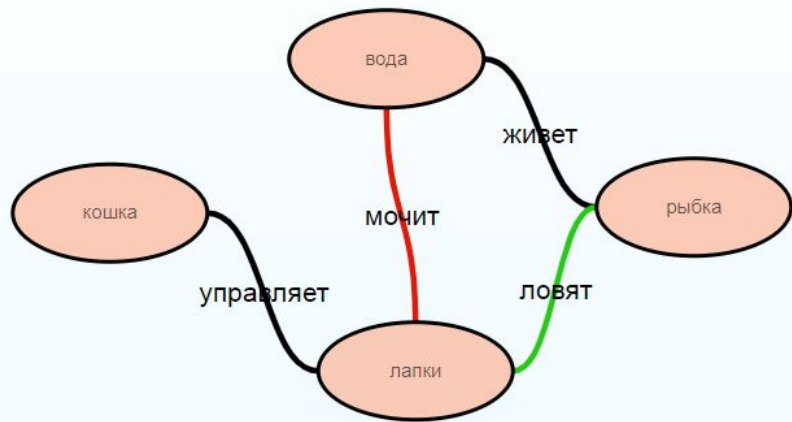
$B2$ - время нормальной работы принтера со времени предыдущей установки нового рулона бумаги.

П1. ЕСЛИ ввести задержку $T=30$ сек. при каждой печати номерка, ТО ОС будет успевать определить, что принтер не работает, НО время ожидания пациентов существенно увеличится.

Решение: опытным путем определить, сколько номерков (N) можно напечатать из одного рулона. Как только количество номерков приближается к N , вводить задержку 30 сек. при печати каждого номерка. Таким образом, система будет работать быстро большую часть времени и замедляться только при приближении к $B1$.

Неправильный граф

В проекте используется библиотека GraphDrakula (GD) на основе RaphaelJS, позволяющая отрисовывать графическое представление графов на веб-странице. Если между двумя вершинами больше одной дуги, GD рисует одну дугу поверх другой, в результате чего “нижняя” дуга не видна. Как быть?

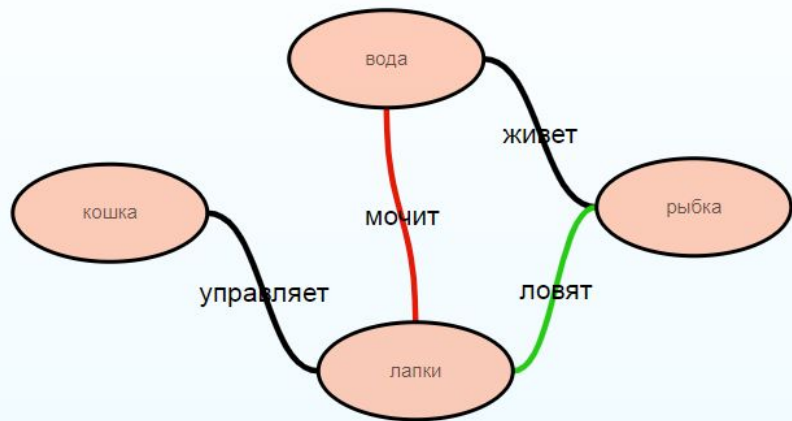


```
/* bezier path */
var x1 = p[res[0]].x,
    y1 = p[res[0]].y,
    x4 = p[res[1]].x,
    y4 = p[res[1]].y,
    dx = Math.max(Math.abs(x1 - x4) / 2, 10),
    dy = Math.max(Math.abs(y1 - y4) / 2, 10),
    x2 = [x1, x1, x1 - dx, x1 + dx][res[0]].toFixed(3),
    y2 = [y1 - dy, y1 + dy, y1, y1][res[0]].toFixed(3),
    x3 = [0, 0, 0, x4, x4, x4 - dx, x4 + dx][res[1]].toFixed(3),
    y3 = [0, 0, 0, y1 + dy, y1 - dy, y4, y4][res[1]].toFixed(3);
/* assemble path and arrow */
var path = ["M", x1.toFixed(3), y1.toFixed(3), "C", x2, y2, x3, y3, x4.toFixed(3), y4.toFixed(3)].join(",");
/* arrow */
if(style && style.directed) {
    /* magnitude, length of the last path vector */
    var mag = Math.sqrt((y4 - y3) * (y4 - y3) + (x4 - x3) * (x4 - x3));
    /* vector normalisation to specified length */
    var norm = function(x,l){return (-x*(l||5)/mag).toFixed(3);};
    /* calculate array coordinates (two lines orthogonal to the path vector) */
    var arr = [
        {x: (norm(x4-x3)+norm(y4-y3)+x4).toFixed(3), y: (norm(y4-y3)+norm(x4-x3)+y4).toFixed(3)},
        {x: (norm(x4-x3)-norm(y4-y3)+x4).toFixed(3), y: (norm(y4-y3)-norm(x4-x3)+y4).toFixed(3)}
    ];
    path = path + "M"+arr[0].x+"", "L"+arr[0].x+"", "L"+arr[1].x+"", "L"+arr[1].x+"";
}
```

Неправильный граф - решение

НАОБОРОТ: дуга становится вершиной

Добавить на дуги вершины нового вида - с описанием дуги. Тогда разные дуги с точки зрения библиотеки будут действительно разными, но для пользователя визуально они будут соединять одни и те же вершины.



```
/* bezier path */
var x1 = p[res[0]].x,
    y1 = p[res[0]].y,
    x4 = p[res[1]].x,
    y4 = p[res[1]].y,
    dx = Math.max(Math.abs(x1 - x4) / 2, 10),
    dy = Math.max(Math.abs(y1 - y4) / 2, 10),
    x2 = [x1, x1, x1 - dx, x1 + dx][res[0]].toFixed(3),
    y2 = [y1 - dy, y1 + dy, y1, y1][res[0]].toFixed(3),
    x3 = [0, 0, 0, x4, x4, x4 - dx, x4 + dx][res[1]].toFixed(3),
    y3 = [0, 0, 0, y1 + dy, y1 - dy, y4, y4][res[1]].toFixed(3);
/* assemble path and arrow */
var path = ["M", x1.toFixed(3), y1.toFixed(3), "C", x2, y2, x3, y3, x4.toFixed(3), y4.toFixed(3)].join(",");
/* arrow */
if(style && style.directed) {
    /* magnitude, length of the last path vector */
    var mag = Math.sqrt((y4 - y3) * (y4 - y3) + (x4 - x3) * (x4 - x3));
    /* vector normalisation to specified length */
    var norm = function(x,l){return (-x*(l||5)/mag).toFixed(3)};
    /* calculate array coordinates (two lines orthogonal to the path vector) */
    var arr = [
        {x: (norm(x4-x3)+norm(y4-y3)+x4).toFixed(3), y: (norm(y4-y3)+norm(x4-x3)+y4).toFixed(3)},
        {x: (norm(x4-x3)-norm(y4-y3)+x4).toFixed(3), y: (norm(y4-y3)-norm(x4-x3)+y4).toFixed(3)}
    ];
    path = path + "M"+arr[0].x+" "+arr[0].y+"L"+x4+" "+y4+"L"+arr[1].x+" "+arr[1].y;
}
```



Передача закладных

В программе на VBScript есть массив ссылок на объекты типа “Закладная”.

Требуется сформировать и отправить на печать документы в MS Word. Каждый документ - это акт передачи закладных исполнителю. В акте передаче можно указывать несколько закладных, если исполнитель у них один.

Исполнитель указан в объекте “Закладная”.

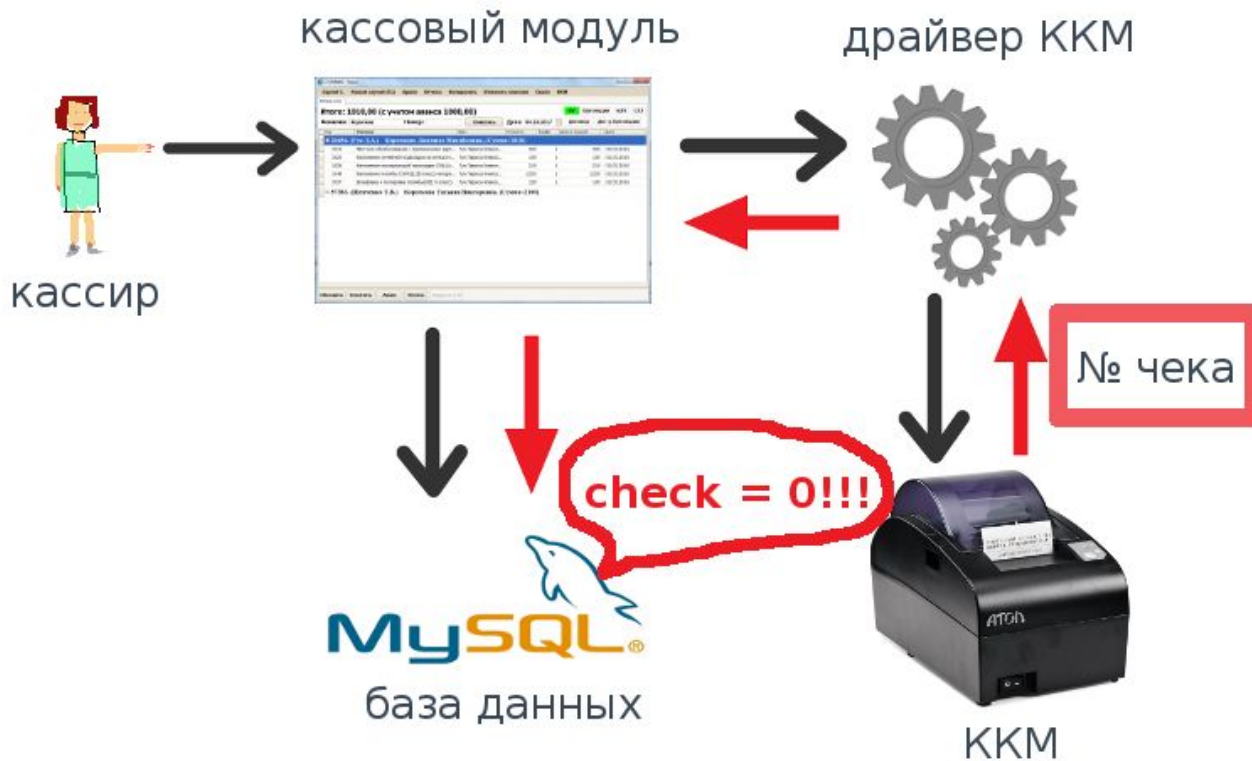
Как это сделать?



Передача закладных - решение

Необходимо сделать сортировку массива закладных по исполнителю. Тогда задача сводится к более простой задаче - формированию акта для исполнителя, когда исполнитель всего один.

Незадачливый кассир





Незадачливый кассир - решение

Задача для диверсионного анализа.

Контрольное решение: кассир открывала 2 экземпляра приложения (открывала 2-й, забыв, что 1-й открыт и свернут). 1-й модуль захватывал драйвер ККМ, в результате чего, 2-й модуль думал, что ККМ не подключен и писал нули в базу прямо в середине рабочей смены.

Ленивые ортопеды



Перед бесплатным протезированием у ортопеда пациент проходит подготовку у терапевта. Услуги оплачивает город, но услуги терапевта будут оплачены только если их утвердит ортопед.

Ортопеды забывают и ленятся утверждать услуги терапевтов. Терапевты теряют деньги.

Заведующие терапевтического и ортопедического отделения близки к войне.

Что можно сделать на уровне МИС?

Ленивые ортопеды - решение



При добавлении терапевтических услуг автоматически выставлять признак того, что они приняты ортопедом. Если ортопед забудет или поленился проверить услуги, то проблемы будут у него, а не у терапевта.



Петина библиотека

Петя решил сделать электронную библиотеку и предложил всем читающим друзьям присылать ему электронные книги.

У Пети 1000 000 друзей, все они любят читать, но, к сожалению, иногда одни и те же книги. В первый же день Пете прислали 1000 000 копий “Трёх мушкетеров”.

Для того, чтобы не засорять свою библиотеку копиями, Петя написал программу, которая при поступлении каждой новой книги проверяет, нет ли такой книги в библиотеке. Проверка двух книг происходит посимвольно, от начала и до конца самой короткой из них. Если какой-то символ не совпадает - книги считаются разными (таково условие Пети).

У Пети в библиотеке уже 100 миллионов книг, каждый день друзья присылают еще 1000. Программа не справляется с проверкой, новые книги стоят в длинной очереди, друзья негодуют.

Как быть?



Петина библиотека - решение

Приемы:

16. ЧАСТИЧНОГО ИЛИ ИЗБЫТОЧНОГО ДЕЙСТВИЯ

24. ПОСРЕДНИКА

Петя, при получении книги, вычисляет хэш от ее текста. Все хэши Петя хранит в отсортированном массиве. Перед добавлением книги Петя проверяет, нет ли хэша новой книги в этом массиве (поиск в отсортированном массиве быстрый - за логарифм от размера массива). Если нового хэша нет, значит книга новая - можно добавлять. Если хэш уже есть, значит книга уже есть с большой вероятностью (зависит от выбранной хэш-функции). В этом случае можно сравнить содержимое книги с той, хэш которой подошел.