# Тема проекта

Меня и мою тему представляет секретарь.

# 2. Актуальность

Уважаемые члены государственной экзаменационной комиссии.

Объем мирового рынка САПР по итогам 2021 года составил $9,4 млрд, увеличившись примерно на $0,6 млрд относительно продаж 2020 года в $8,8 млрд. К концу 2021 года в мире насчитывалось около 7 млн пользователей САПР.

Анализ активности пользователей САПР является одним из важных этапов повышения эффективности проектирования, которое необходимо для ускорения развития экономики и промышленности Российской Федерации.

Кроме этого, много отечественных пользователей используют зарубежное ПО, поэтому важно не просто копировать его, а модернизировать и переходить от импортозамещения к импортонезависимости.

Для улучшения качества ПО и эффективности взаимодействия пользователя с системой необходимо проанализировать активность пользователей.

Анализ активности пользователей САПР можно применить для решения следующих задач:

для решения задач представленных на экране

Повышения эффективности использования системы: анализ активности пользователей САПР позволяет разработчикам понять, как пользователи взаимодействуют с системой и внести улучшения в интерфейс или функциональность, для повышения эффективности использования системы.

Оптимизация процесса разработки: на основе полученной информации можно сделать выводы на каких аспектах системы стоит сосредоточить усилия.

А также, выявление проблем взаимодействия пользователя с интерфейсом и ошибок в системе.

Оптимизация пользовательского опыта  
Улучшение процесса разработки  
Выявление необычного или нежелательного поведения

Формирования модели пользовательской активности  
Вывод рекомендаций по улучшению пользовательского интерфейса.  
Создание адаптивной модели пользователя

# 3. Цель и задачи

На следующем слайде представлены цель и задачи.

# 4. Сравнение рассмотренных методов

В ходе работы мною были рассмотрены различные методы анализа активности пользователей, сравнение которых представлено в таблице на слайде. В качестве критериев были выделены: требования к входным данным, учет времени совершения транзакций и сложность алгоритма.

Активность пользователей САПР представляет собой последовательность команд. Например, для простых САПР это могут быть: начертить линию, окружность, дугу, отменить действие и так далее. Каждая команда выполняется в определенный момент времени поэтому за основу был выбран алгоритм GSP, т.к. он учитывает время совершения транзакций.

Поскольку активность пользователей представляет собой последовательность действий, производящихся в определенный момент времени, то за основу был выбран алгоритм GSP, т.к. он учитывает время совершения транзакций.

# 5. Формализованная постановка задачи

Задача анализа активности пользователей САПР представлена в нотации IDEF0, которую Вы можете видеть на слайде. На вход программе подаются информация о выполненных командах и пользовательские параметры: минимальный уровень поддержки, минимальный и максимальный временные разрывы между командами в секундах. Ограничения на входные данные представлены на экране

Используя методы поиска последовательных шаблонов, система определяет часто встречающиеся последовательности команд, их уровень поддержки и коэффициент зависимости.

# 6. Уровень поддержки

Значение поддержки последовательности равно проценту сессий, которые ее содержат.

За сессию будем считать все команды, выполненные до завершения работы пользователя САПР, либо до переключения на работу с новым файлом.

Сессия содержит последовательность, если в ней есть все элементы в том же порядке и временные промежутки между каждыми двумя соседними командами удовлетворяют параметрам, заданным пользователем.

На слайде представлен пример расчета поддержки для последовательности <1,2> при четырех заданных сессиях и допустимом временном разрыве между командами от 0 до 2 секунд. Поскольку последовательность <1,2> содержится в 3ех из 4ех сессиях, то ее поддержка равна 0.75.

# 7. Коэффициент зависимости

Коэффициент зависимости показывает, на сколько команды в последовательности зависят друг от друга и считается как отношение поддержки последовательности к произведению поддержек всех ее подпоследовательностей, состоящих из 1 команды. Если значение коэффициента <= 1, значит зависимости нету. Если же > 1, то зависимость есть. Чем больше единицы, тем вероятней то, что эти команды использовались вместе.

На слайде представлен пример расчета коэффициента зависимости.

На слайде представлен пример расчета коэффициента зависимости для последовательности <1, 2, 3>, поддержка которой равна 0.5, а поддержки ее подпоследовательностей равны 0.6, 0.8 и 1 соответственно.

В таком случае, коэффициент зависимости примерно равен 1.041, а значит есть зависимость между командами в этой последовательности.

# 8. Генерация последовательностей

Разработанный метод состоит из двух основных этапов: Генерация последовательностей и Подсчет их поддержки. На следующих слайдах представлены схемы для данных этапов.

При генерации просто перебираются все сочетания текущих последовательностей, и если условие их объединения выполняется, то добавляется новая последовательность.

# 9. Проверка поддержки последовательности сессией

При подсчете поддержки каждая сессия проверяется на содержание рассматриваемой последовательности. На данном этапе алгоритм переключается между двумя фазами: поиск следующей и предыдущей команды (по англ. forward и backward phase). Происходит это пока последовательность не будет полностью найдена или же какая-либо команда из нее не будет отсутствовать.

# 10. Структура программного обеспечения

На следующем слайде представлена структура ПО.

Модуль преобразования логов записывает данные в таблицу с помощью модуля взаимодействия с базами данных. А модуль вычисления часто встречающихся последовательностей, реализующий разработанный метод, на основе этой базы данных вычисляет результат, для передачи его пользователю через интерфейс.

# 11. Сравнительный анализ времени выполнения этапов метода

После разработки ПО мною был проведён сравнительный анализ времени выполнения этапов метода. По графику видно, что подсчет поддержки последовательностей занимает большую часть времени, чем их генерация.

# 12. Сравнительный анализ времени выполнения метода в зависимости от параметров

Также мной был проведен сравнительный анализ времени выполнения разработанного метода в зависимости от его параметров.

При анализе использовались данные логов разработчиков САПР nanoCAD, разного размера: 60, 30, 10 и 1 тысяча команд.

По 1ому графику видно, что при уменьшении минимального уровня поддержки, время выполнения будет расти т.к. в таком случае больше последовательностей будут проходить отбор.

На 2ом графике мы видим, что при увеличении минимального временного разрыва между командами, время выполнения уменьшается т.к. в таком случае получается меньше последовательностей из-за увеличения ограничения.

На последнем графике видно, что при уменьшении максимального временного разрыва между командами, время выполнения тоже уменьшается, потому что в этой ситуации, также получается меньше последовательностей проходят ограничения.

Из проведенного исследования можно сделать вывод, что если необходимо уменьшить время выполнения, то в результате будет меньше последовательностей. Поэтому параметры следует подбирать в зависимости от того, что в приоритете для решаемой задачи, скорость или информативность.

Например, для предсказания следующей команды пользователю, необходима скорость, чтобы программа была отзывчивой, а в случае анализа активности исследователем, без ограничения во времени, важнее будет информативность.

# 13. Заключение

По итогу проделанной работы была достигнута цель и решены все поставленные задачи, перечисленные на экране.

По итогу проделанной работы была достигнута цель – разработан и программно реализован метод анализа активности пользователей САПР с использованием поиска последовательных шаблонов, а также были решены все поставленные задачи, перечисленные на экране.

# 14. Перспективы дальнейшего развития

В качестве возможных путей развития данной разработки можно выделить следующие:

* Возможность предсказания следующей команды пользователю, что улучшит эффективность взаимодействия с системой
* Подсчет процента содержания найденных последовательностей в сессиях или
* Вычисление времени, необходимого для выполнения последовательности для повышения информативности результата
* Создание адаптивной модели пользователя

# Демонстрация

Спасибо за внимание, готов ответить на ваши вопросы и перейти к демонстрации ПО.

Для демонстрации используются логи в которых большинство сессий выполнялась одна и та же последовательность команд: линия, дуга, круг и отмена. Запустим программу с минимальным уровнем поддержки 0.5, минимальным и максимальным разрывами 0 и 15 секунд соответственно. В результате появляется та самая последовательность, которая часто встречается в данных логах. К тому же, чем полноценней последовательность, тем выше коэффициент зависимости, таким образом можно определить, какая из последовательностей больше всего походит на шаблон поведения пользователя.

Демонстрировать на показательном примере

Сделать видео демонстрации (секунд 15)