



АГР

Автоматизация
Геологоразведочных Работ

АГР «Консоль бурильщика»
(программа для ведения
электронных буровых журналов на
мобильных устройствах)

Поддержание жизненного цикла

Общие сведения

Программа для ведения электронных буровых журналов на буровых установках в полевых условиях на мобильных устройствах (**АГР «Консоль бурильщика»**) является коммерческим продуктом, разработанным и поддерживаемым профессиональным разработчиком программного обеспечения – компанией ООО «АГР Софтвер» (компания является российской инновационной ИТ-компанией, входит в реестр аккредитованных ИТ-компаний и зарегистрирована в реестре участников «Сколково» за №1121237, <http://sk.ru/net/1121237/>).

Программа **АГР «Консоль бурильщика»** обеспечена необходимыми техническими инструментами и инфраструктурой для того, чтобы конечные потребители (в основном недропользователи и буровые компании) могли эффективно использовать программу для решения производственных задач (сопровождение буровых работ). Программа применима для сопровождения любых видов бурения (колонковое, шарошечное, шнековое, пневмоударное, РС, ударно-канатное и пр.) для решения широкого круга геологических задач (геология, геотехника, гидрогеология, инженерная геология и др.) на разных стадиях освоения объектов (поиски, разведка, доразведка, эксплуатация).

Программа **АГР «Консоль бурильщика»** обеспечена профессиональной технической поддержкой на территориях РФ, Казахстана, Кыргызстана, Армении, Грузии и в других странах.

Разработка и дальнейшее совершенствование Программы **АГР «Консоль бурильщика»** осуществляется в строгом соответствии с версионным регламентом, что позволяет выпускать на рынок качественное и востребованное отечественное решение для автоматизации задач, связанных с выполнением буровых работ.



Сборка приложения и тестирование

Сборка приложения реализована с помощью системы сборки Gradle. Эта система доступна как для эксплуатации в рамках среды разработки отдельным разработчиком, так и широко используется для автоматизации производственных процессов разработки с использованием специализированных серверов и дополнительного программного обеспечения, отслеживающего изменения в репозиториях исходного кода и запускающего сборку ПО после получения данных о новых версиях исходного кода.

При разработке учитывалась необходимость в автоматизированном тестировании функционала приложения. За счёт того, что большая часть функционала запускается в обычном браузере, можно использовать средства автоматизированного тестирования наподобие Selenium. При запуске приложения в браузере исключается только функциональность, касающаяся нативного кода, взаимодействующего с ОС Android (только работа с Bluetooth по передаче изменений с планшета на другие устройства). Но эта функциональность все равно может быть проверена только вручную, после сборки и запуска приложения на устройстве. Остальной же функционал работает в штатном режиме, что позволяет проводить тестирование бизнес-логики приложения. Для организации подобного тестирования в приложении разработано специализированный API, позволяющий запускать тот или иной функционал приложения через скриптовую подсистему среды автотестирования.

Процесс разработки ПО происходит по фактическому адресу: 196247, г. Санкт-Петербург, Ленинский проспект, дом 153, офис 501 и 502.

В разработке задействована команда из 11 человек. Все члены команды имеют достаточную квалификацию и опыт работы, для выполнения своих функций. Команда состоит из: Руководителя проекта, Технического директора, Системного аналитика, Ведущего инженера-геолога, Горного инженера, двух Инженеров-программистов 1 категории, двух инженеров-программистов 2 категории и двух тестировщиков программного обеспечения 2 категории.

Обеспечение технической поддержкой

Техническая поддержка программного комплекса АГР 4 осуществляется следующими уполномоченными компаниями:

ООО «АГР Софтвр» - разработчик и правообладатель программы АГР «Консоль бурильщика». Офис компании находится в городе Санкт-Петербург (Почтовый адрес: 196247, г. Санкт-Петербург, Ленинский проспект, д.153, лит.А, офис 501). Процесс сопровождения ПО осуществляется по адресу: 196247, г. Санкт-Петербург, Ленинский проспект, д.153, лит.А, офис 501 и 502.

Техническая поддержка осуществляется следующими способами:

- По электронной почте support@agrsoftware.ru;
- По телефону +7 812 384-48-09 часы работы по МСК 09:00-18:00;
- Выезд специалиста ООО «АГР Софтвр» на объект заказчика.



ООО «АГР Софтвр»

Автоматизация Геологоразведочных Работ

www.agrsoftware.ru

Применяемые термины и определения

Техническая поддержка – совокупность услуг, оказываемая Подрядчиком Заказчику в соответствии с условиями договора технической поддержки и включающая в себя:

- консультирование **уполномоченных сотрудников** Заказчика, использующих программу АГР «Консоль бурильщика» при исполнении своих должностных обязанностей по вопросам, связанным с работой программы АГР «Консоль бурильщика», в том числе по структуре, таблицам, внутреннему наполнению АГР (включая консультации по поводу установки обновлений), ведения буровых журналов;
- обновление программы АГР «Консоль бурильщика».

Уполномоченный сотрудник Заказчика – работник Заказчика, работающий с программой АГР «Консоль бурильщика» на предприятии Заказчика.

Запрос – обращение Заказчика (**уполномоченного сотрудника** Заказчика) за **технической поддержкой**, по телефону или обращение, оформленное в соответствии и по форме договора технической поддержки (форма приведена ниже).

Краткая характеристика запроса	Приложение № 4
Категория*:	к Договору № _____ от 09.01.2023
Приоритет**:	
Объект:	Бланк Запроса на оказание Технической поддержки
ФИО:	
Телефон:	
Эл. почта:	

З А П Р О С

Заказчик _____, в соответствии с Договором на техническую поддержку предлагает Подрядчику, ООО «АГР Софтвр», рассмотреть и согласовать настоящий Запрос:

К настоящему Запросу прилагаются следующие материалы***:

- _____
- _____

Представитель Заказчика, _____ 20__ г.
действующий на основании _____

_____ (_____)
(должность) (подпись) (ФИО)

Услуги оказаны в соответствии с Запросом _____ 20__ г.
Заказчик _____ Подрядчик

_____ (_____)
(подпись) (ФИО) (подпись) (ФИО)

* - в поле «категория» подателем указывается обобщенный тип изложенной в заявке проблемы (ошибка в программе, ошибка в шаблоне, вопрос по программе, и т.п.)

** - в поле «приоритет», в соответствии со срочностью необходимого решения изложенной проблемы, указывается приоритет запроса, в виде одного из следующих значений, в зависимости от излагаемой проблемы:

CRITICAL (критический) - оказывает критическое влияние на производственный процесс или подвергает его риску.

URGENT (срочный) - оказывает значительное влияние на производственный процесс, но не препятствует его выполнению.

NORMAL (обычный) - оказывает незначительное влияние на производственный процесс без заметного ущерба.

LOW (низкий) - Проблема не оказывает какого-либо влияния на производственный процесс. Типичными запросами такого приоритета являются запросы вида «Баг, cr...», а также относящиеся к API и вопросам совместимости, инсталляции и т.п.

*** - перечислить и приложить документы, при необходимости передачи последних для более полной характеристики проблемы. Обязательно указывается способ их передачи. Например: «1. Файл скважины (номер) в виде файла формата agr, высылаемого по электронной почте.» При отсутствии дополнительно передаваемых данных, указать «материалы отсутствуют».



Запрос пользователя в службу технической поддержки с проблемой по работе приложения попадает в систему обработки запросов. Для каждого запроса создается отдельный баг-репорт, который содержит информацию о проблеме, ее описании, шагах для воспроизведения проблемы и деталях, при которых произошла проблема. После создания баг-репорта каждому запросу присваивается уникальный идентификатор, который используется для отслеживания статуса запроса.

Следующий шаг в обработке запроса — это его классификация. Запросы классифицируются по степени их сложности и важности. Это позволяет определить приоритет запроса и скорость его обработки. Запросы со статусом ошибки CRITICAL, будут отработаны в первую очередь, следующим по приоритету идет URGENT. Дальше отрабатываются запросы со статусом NORMAL и в последнюю очередь попадает приоритет LOW.

Срок отработки запроса может варьироваться в зависимости от сложности запроса и степени его важности. Время отработки запроса технической поддержкой может составлять от нескольких минут до нескольких часов. Если запрос связан с проблемой, которая может потребовать изменения исходного кода приложения, то время отработки может быть довольно длительным.

Версионность АГР

Программа АГР «Консоль бурильщика» постоянно совершенствуется. Производителем несколько раз в год выпускаются новые версии программы, а раз в 3-4 года выходит новое поколение системы.

Нумерация версий программы АГР «Консоль бурильщика» идет в строгом соответствии с регламентом, приведенном ниже.

Формат нумерации версий программы АГР «Консоль бурильщика».

Формат версии выглядит как – X–Y–Z где:

- X – обозначает поколение программы. Номер поколения является важным маркетинговым инструментом, позволяющим позиционировать комплекс на рынке и обозначать его технологический уровень развития. С этим номером также связан учет программных продуктов АГР как объектов НМА (нематериальные активы).
- Y – основной номер релиза (мажорный номер версии). Меняется в среднем 2-3 раз в течение года. Изменение номера релиза обычно связано с добавлением нового функционала. Следует заметить, что изменения, подразумевающие изменение номера релиза, не являются слишком существенными, например, такие изменения не затрагивают пересмотр архитектуры программного продукта.
- Z – дополнительный номер релиза (минорный номер версии). Может меняться достаточно часто, например, 1-2 раза в месяц. Обычно изменение номера вызвано обратной связью пользователей АГР, информирующей о тех или иных



недоработках или неудобствах в новом функционале, что приводит к необходимости доработки выпущенной версии и созданию исправлений.

Технический номер версии

В процессе разработки номер версии дополняется рядом полей, обозначающих номер Release Candidate. Это так называемый технический номер версии, существующий только в течение времени тестирования новой версии продукта. В последующем этот номер становится неактуальным. Технический номер версии имеет формат аналогичный основному, за исключением того, что к номеру добавлено одно поле. Иными словами, формат технической версии выглядит как – X-Y-Z-AAAA где "AAAA" номер ревизии из системы версионного контроля. Технические версии не передаются заказчиком. С этими номерами версий работают только собственные подразделения компании, вовлеченные в процессы подготовки нового релиза к выпуску.

Регламент присвоения номера поколения программы АГР

Присвоение нового номера поколения выполняется, когда вносятся радикальные изменения в архитектуру программного комплекса, что приводит к невозможности обратной совместимости между вновь выпускаемыми и предыдущими версиями продуктов (например, в старой версии уже нельзя открыть документы создаваемые в новой версии и уже невозможно решить проблему совместимости выпуском исправлений для более ранних версий)

Инкремент версии равен 1(единице). Разработчики используют этот номер для каталогизации версий исходного кода, каталогизации User Story, Bug Tracking и т.д (применяется Scrum-технология разработки ПО). При изменении номера поколения, материалы при необходимости дублируются в хранилищах нового поколения, а все материалы по предыдущему поколению переводятся в разряд архивных и используются только для ретроспективы.

Регламент присвоения основного и дополнительного номеров версии программы АГР «Консоль бурильщика» (см. табл.2)

Таблица 2. Регламент присвоения основного и дополнительного номеров версии.

#	Этап	Время (рабочие дни)	Описание
1.	Определить перечень дорабатываемого функционала.	~10	Проводится ряд совещаний, на которых обсуждается необходимость разработки или переработки текущего функционала. Разработчики создают User Story, включающую все задачи по доработке функционала. User Story получает номер версии. В свою очередь в системе версионного контроля создается Branch соответствующий номеру версии и имеющий суффикс -alpha например -



			<p>4.1.0-alpha, являющейся копией Trunk. В контексте предыдущей версии теперь возможен выпуск только исправлений. По завершению этапа появляется согласованная и утвержденная User Story с номером, соответствующим новому номеру релиза. То есть, происходит увеличение параметра Y или, другими словами, увеличивается основной номер релиза.</p> <p>Объем работ, определяемый для User Story целесообразно определять из расчета длительности фазы разработки не более 20 рабочих дней (фактически календарный месяц).</p>
2.	Доработать существующий функционал	~20	<p>Как уже было отмечено, все работы по предыдущим версиям, ведутся в соответствующих Branches. Работа над функционалом ведется непосредственно в Branch активной User Story. По завершению работ соответствующая User Story ветка переименовывается в соответствующее номеру версии название, но с суффиксом -beta. Например - 4.1.0-beta. Версии продуктов получают номера в соответствии с номером версии и номером ревизии текущего Branch, в котором ведется работа. Например, 4.1.0.XXXX, где XXXX номер ревизии Branch из системы версионного контроля. С beta ветками могут работать инженеры по тестированию.</p>
3.	Тестировать доработанный функционал	~10	<p>По завершению доработки выполняется сборка Release Candidate. В процессе тестирования, обнаруженные ошибки и недоработки фиксируются в соответствующем разделе системы Bug Tracking, принимаются и обрабатываются разработчиками. В конце каждого рабочего дня все завершенные исправления фиксируются в системе версионного контроля, а утром следующего дня собирается очередная версия Release Candidate. Таким образом, номера технической версии представляют собой следующее - 4.1.0.8945, следующий день 4.1.0.8948 и т.д. Инкремент версии соответствует числу фиксаций исходного кода в системе версионного контроля в соответствующей ветке.</p>
4.	Выпустить релиз	~2	<p>После того как все приложения проверены, все ошибки и недоработки, которые можно устранить, исправлены, разработчики выпускают релиз. Номер технической версии становится неактуальным, а продукты получают окончательный номер релиза. Например, 4.1.0. После чего, для релиза, на основании User Story связанной с ним, формируется список изменений, сопровождающих релиз, собираются установщики, при необходимости вносятся изменения в лицензию ключей и выполняется их обновление у клиентов, а затем установщики становятся доступными для рассылки клиентам. Ветка в системе версионного контроля, в которой велась работа над выпущенной версией, переименовывается в -stable и затем на основе этой ветки в Tags создается ветка, соответствующая номеру релиза, например, 4.1.0.</p>



5.	Принимать сообщения об обнаруженных ошибках или недоработках	~40	В течение не более 10 рабочих дней после выпуска релиза, могут быть получены и обработаны замечания или ошибки, обнаруженные непосредственно пользователями во время эксплуатации новой версии продуктов. Полученные ошибки и замечания фиксируются в системе Bug Tracking в разделе, соответствующем текущей версии продукта.
6.	Исправить ошибки	~40	<p>В зависимости от приоритета ошибки или недоработки к её решению приступают либо сразу, либо фиксируют в User Story для следующей версии продукта. Сразу рассматриваются только те ошибки или недоработки, которые приводят к тому, что с помощью продукта пользователи не могут выполнять свои служебные обязанности, то есть ошибки, создающие прямую угрозу производственным процессам клиентов. При выпуске исправления изменяется дополнительный номер релиза. Например, для релиза за номер 4.1.0 номера релизов содержащих те или иные исправления выглядят так - 4.1.1, 4.1.2 и т.д. Перед началом работ в системе версионного контроля создается Branch с номером, соответствующим номеру версии с исправлениями. Например, если ошибка обнаружена в релизе 4.1.0, значит будет создан Branch с номером 4.1.1. И в нем разработчики выполняют работы по решению проблемы.</p> <p>Так же, как и в процессе разработки, ежедневно осуществляется фиксация изменений в системе версионного контроля, а на следующее утро выполняется сборка нового релиза, который получает технический номер релиза, например 4.1.1.8990. С этим релизом работает только отдел тестирования.</p>
7.	Выпустить исправление	~1	По завершению процесса исправления разработчики выпускают исправление. Инкремент дополнительного номера составляет 1(единицу). Категорически запрещено изменять номер на большее значение, например ситуация 4.1.1 - > 4.1.4 недопустима. В системе версионного контроля в Tags фиксируется ветка - копия с ветки текущего сопровождаемого релиза и ей присваивается номер с увеличенной на 1-у дополнительной версией продукта. Например, 4.1.0: исправление 1 - 4.1.1, следующее будет 4.1.2 и т.д.