

Процесс обеспечения качества

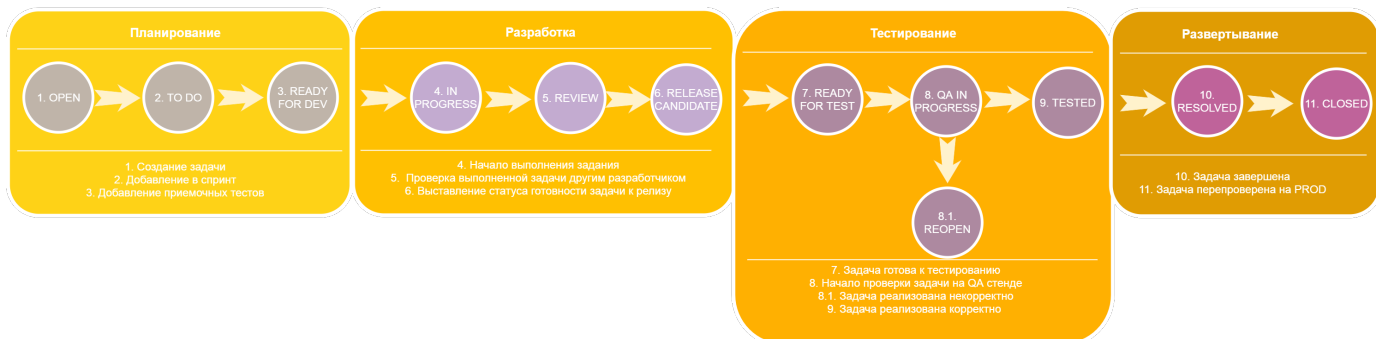
- Проведение тестирования на каждом этапе жизненного цикла ПО
- Жизненный цикл задачи
- Работа с дефектами в JIRA
- Test management system
- Критерии
- Статический анализ кода
- Регрессионное тестирование
- Тестирование API
- Исследовательское тестирование
- Интеграционное тестирование
- Тестирование отказоустойчивости
- Приемочное тестирование
- Тестирование производительности, нагрузочное тестирование, стресс-тестирование
- Тестирование инфраструктуры
- Тестирование совместимости
 - Кроссбраузерное
 - Открытие/редактирование отчетов
- Автоматизированное тестирование
 - Схема 1. Процесс автоматизации тестирования
 - Схема 2. Детальная схема внесения новых изменений в develop ветку
 - Инструменты
- Проверка пользовательских сценариев перед демо
- Мониторинг
- Деплой
- Чек-лист тестов перед релизом
- Полезные ссылки

Проведение тестирования на каждом этапе жизненного цикла ПО

Этапы	Комментарии
Планирование	
Составление приемочных тестов по спецификациям в Test Management	Как правило позитивные приемочные тесты
Прикрепление тест кейсов к задачам на разработку и перевод задачи в статус READY FOR DEV	
Убеждение, что все необходимые задачи/подзадачи созданы	Бывают случаи, когда на бэк задача есть, а на фронт не создана
Разработка	
Дополнение негативными тестами в Test Management	
Автоматизация тестовых сценариев по приоритету	
При принятии задачи на тестирование проверка того, что задача в статусе READY FOR TEST, все прикрепленные тест кейсы к задаче успешно пройдены на DEV стенде и функционал в нужной нам ветке	Если задача техническая, то должно быть описание, написанным разработчиком, того как тестировать
Тестирование	
Обновление тестового стенда после отрезания релизной ветки	При необходимости привлекаются девопсы и разработчики
Тестирование нового функционала <div><div>Функциональное тестирование через API, UI</div><div>Проверка удобства использования в BackOffice, FrontOffice, CallCenter</div></div>	Задача в статусе READY FOR TEST

Тестирование производительности	
Тестирование обновления приложения под нагрузкой	
Стресс тестирование - для выявления текущего статуса работы приложения под нагрузкой	
Тестирование стабильности - для выявления утечек памяти и других проблем, связанных с длительностью работы системы	
Проведение регресса	Выполнение полной регрессии по тестовому набору. Запускаются все автотесты, так же проведение мануального тестирования на неавтоматизированные сценарии
Эксплуатация (развертывание)	
Перепроверка задач в статусе Tested на боевом окружении	Есть риск того, что некоторые настройки будут забыты примениться на боевом окружении и функционал не будет работать
Поддержка	
При возникновении непредвиденных ошибок может потребоваться вмешательство QA	
Проблемы с сетью	
Проблемы интеграции с другой системой	
Инфраструктурные проблемы	

Жизненный цикл задачи



Работа с дефектами в JIRA

- При заведении бага необходимо добавить информацию:
 - Шаги и действия, с помощью которых можно воспроизвести ошибку. Скриншоты, логи, трейсы при необходимости
 - На каком стенде обнаружен
 - Priority
 - Версия
 - Номер спринта, в котором баг обнаружен
 - Label, components
- При закрытии бага:
 - В какой версии был проверен дефект
 - На каком стенде, сервере
 - Комментарий, дополнительная информация в виде скриншотов, логов (при необходимости)
- Дефекты с высшим приоритетом добавлять в регрессионный набор тестов

Test management system

Для тест дизайна используется плагин к JIRA - Adaptavist Test Management for JIRA. Данные по тесткейсам, требованиям и результатам ручного тестирования хранятся в JIRA, вкладке Tests. Результаты автотестов в [Jenkins](#).

Выбор инструмента для управления требованиями.

Критерии

Приёмочные критерии (acceptance criteria) — любые объективные показатели качества, которым разрабатываемый продукт должен соответствовать, чтобы считаться готовым к переводу на полноценное тестирование.

Критерии начала тестирования (entry criteria) — перечень условий, при выполнении которых команда приступает к тестированию. Наличие этого критерия страхует команду от бессмысленной траты усилий в условиях, когда тестирование не принесёт ожидаемой пользы.

Критерии приостановки тестирования (suspension criteria) — перечень условий, при выполнении которых тестирование приостанавливается. Наличие этого критерия также страхует команду от бессмысленной траты усилий в условиях, когда тестирование не принесёт ожидаемой пользы.

Критерии возобновления тестирования (resumption criteria) — перечень условий, при выполнении которых тестирование возобновляется (как правило, после приостановки).

Критерии завершения тестирования релизной ветки (exit criteria) — перечень условий, при выполнении которых тестирование релизной ветки завершается. Наличие этого критерия страхует команду как от продолжения тестирования в условиях, когда оно уже перестаёт приносить ощутимый эффект.

Приёмочные критерии	Критерии начала тестирования	Критерии приостановки тестирования	Критерии возобновления тестирования	Критерии завершения тестирования релизной ветки
Успешное прохождение 100% тест-кейсов на DEV стенде прикрепленных к задаче (новому функционалу)	Выход релизной версии	Если при выполнении не менее 50% запланированных тест-кейсов более 50% из них завершились обнаружением дефекта	Исправление более 50% обнаруженных на предыдущей итерации дефектов	Выполнение более 80% запланированных на итерацию тест-кейсов
Итоговое покрытие требований тест-кейсами должно составлять не менее 80%	QA стенд работает без инфраструктурных ошибок	При возникновении инфраструктурных проблем	При исправлении инфраструктурных проблем	По истечении времени

Статический анализ кода

Используется SonarQube. Разработчики правят алерты.

Считается ли покрытие кода юнит-тестами? jасосо какой-нибудь

Регрессионное тестирование

В идеале проводить для каждой сборки весь набор автотестов. Если тесты будут долго проходить (е2е-тесты обычно долгие), то параллелить.

Тесты на UI проводить для всех браузеров, использовать Selenium Grid и несколько виртуалок с браузерами. Перед релизом проводить полное регрессионное тестирование, включая мануальные тесты.

✓ [Сценарии для регрессионного тестирования](#)

Основные тестовые наборы/кейсы для проведения регресса

Подписки		Приоритет	Обязательная проверка в браузерах по приоритетам				TM link
WAP подписка	Успешная подписка	High	IE	Chrome	Firefox	Safari	T117
	Продление	High					T95
	Тгу подписка	Normal					T310
WEB подписка	Успешная подписка	High	IE	Chrome	Firefox	Safari	T128
	Продление	High					T487
	Тгу подписка	Normal					
SMS подписка	Успешная подписка	High					T120
							T119
	Продление	High					T480
	Тгу подписка	Normal					T309
							T303
USSD подписка	Успешная подписка	High					T118
	Продление	High					T486
	Тгу подписка	Normal					
Generate подписка	Успешная подписка	Normal	IE	Chrome	Firefox	Safari	T123
	Продление	Normal					T485
	Тгу подписка	Normal					T311
Send_code/act подписка	Успешная подписка	High					T124
	Продление	High					T482
	Тгу подписка	Normal					T315
Subscription	Успешная подписка	High					T13
	Продление	High					T481
	Тгу подписка	Normal					T313
PUSH подписка	Успешная подписка	High					T121
	Продление	High					T484
	Тгу подписка	Normal					
СУПП API	Успешная подписка	High					T12
	Продление	High					T483
	Тгу подписка	Normal					T316
Покупки							
WAP покупка	Успешная покупка	High	IE	Chrome	Firefox	Safari	T127
WEB покупка	Успешная покупка методом buy	Low					T130
							T129
	Успешная покупка методом partnerbuy	High					T133
							T134
	Успешная покупка методом fund	High					T135
SMS покупка	Успешная покупка с AoC	High					T175
							T200
							T21
							T194
							T19
	Успешная покупка без AoC	High					T22
USSD покупка	Успешная покупка без меню	High					T202
							T34
							T205
	Успешная покупка с меню	High					T207
Generate покупка	Успешная покупка	High	IE	Chrome	Firefox	Safari	T106

Отписки							
SMS отписка	Успешная отписка	High					T321
USSD отписка	Успешная отписка	Low					T122
LP отписка	Успешная отписка	Low					T322
SUBSCRIPTION отписка	Успешная отписка	High					T323
SEND_CODE/ACT отписка	Успешная отписка	High					T320
Отписка через BO	Успешная отписка с уведомлением	Normal					T630
	Успешная отписка без уведомления	Normal					T324
	Успешная отписка абонентов из файла	Normal					T356
Отписка через КЦ	Успешная отписка	High	IE	Chrome	Firefox	Safari	T325
	Успешная отписка абонентов из файла	High	IE	Chrome	Firefox	Safari	T357
СУПР API отписка	Успешная отписка из личного кабинета	High					T343
	Успешная отписка из мобильной версии личного кабинета	High					T346
Отписка через портал моих подписок	Успешная отписка	High					T347
Отписка при миграции с услуги на услугу	Успешная отписка	Normal					T348
Автоотписка	Автоотписка при запрете	High					T353
							T354
	Автоотписка при неуспешном продлении партнерской услуги	Normal					T351
	Автоотписка при неуспешном продлении собственной услуги	Normal					T352
	Автоотписка при статусе услуги "Отключена"	Normal					T355
Запреты							
Срочные запреты	Установка срочного запрета через SMS	High					T57
							T61
							T65
	Установка срочного запрета через BO	Normal					T467
							T468
	Добавление абонента в черный список из BO						T470
	Установка срочного запрета через КЦ (файл)	High	IE	Chrome	Firefox	Safari	T369
	Снятие абонента из черного списка						T476
	Снятие срочного запрета через BO	Normal					T475
	Снятие срочного запрета через КЦ (файл)	High	IE	Chrome	Firefox	Safari	T370
Бессрочные запреты	Установка бессрочного запрета через BO	Normal					T465
	Снятие бессрочного запрета через BO	Normal					T471
Premium-Rate	Установка запрета PremiumRate на CPA услуги	High					T195
	Снятие запрета PremiumRate из CPA услуг	High					T196
Интерфейс BO							

Услуги	Создание услуги по WAP подписке	Normal	IE	Chrome	Firefox	Safari	T326
	Создание услуги по WEB подписке	Normal					T328
	Создание услуги по WAP покупке	Normal					T330
	Создание услуги по WEB покупке	Normal					T332
	Создание услуги по SMS подписке (партнерская)	Normal					
	Создание услуги по SMS подписке (собственная)	Normal					T338
	Создание услуги по USSD подписке	Normal					
	Создание услуги по SMS покупке	High					T243 T199 T242 T238
	Создание услуги по SMS MT покупке	Normal					
	Создание услуги по USSD покупке	High					T244 T241
	Редактирование услуги партнерской	Normal					T334
	Редактирование услуги собственной	Normal	IE	Chrome	Firefox	Safari	
Шаблоны	Создание шаблона по WAP подписке	Normal	IE	Chrome	Firefox	Safari	T372 T374 T376
	Создание шаблона по WEB подписке	Normal					T377 T379
	Создание шаблона по WAP покупке	Normal					T381 T382
	Создание контент провайдера	Low					T386
	Создание SMPP аккаунта	Low					T251 T254 T255
Интерфейс FO							
Услуги	Создание услуги по WAP подписке	Normal					
	Создание услуги по WEB подписке	Normal					
	Создание услуги по SMS подписке (партнерская)	Normal	IE	Chrome	Firefox	Safari	
	Создание услуги по USSD подписке	Normal					
	Редактирование услуги	Normal	IE	Chrome	Firefox	Safari	
Шаблоны	Создание шаблона по WAP подписке	Normal					
	Создание шаблона по WEB подписке	Normal	IE	Chrome	Firefox	Safari	
	Создание шаблона по WAP покупке	Normal					
Интерфейс КЦ							
Управление учетными записями	Создание учетной записи	High	IE	Chrome	Firefox	Safari	T389
	Редактирование учетной записи	High	IE	Chrome	Firefox	Safari	T390

Радар							
	Подписка через Subscription метод	High					T300
	Изменение канала при подписке из API на WAP	High					T302
Отчеты							
PSM	Финансовая статистика	Normal					
	GeneralReport	Low					
	GeneralReport по СУ	Normal					
	Агрегированный GeneralReport	Normal					
	Отчет по отпискам и запретам	Low					
	Детальный отчет по транзакциям	Normal					
	Еженедельный отчет по партнерским подпискам	Normal					
CPA	Детализированный отчет	Normal					T286
							T287
							T291
							T294
							T450
Порталы							
Портал "Мои подписки"		Normal					
Портал "Мои покупки"		Normal					

Тестирование API

Необходимо добавить тесты на валидацию API. (<https://habrahabr.ru/company/2gis/blog/186768/> - вот пример того, как и для чего это сделано). Разработчикам фронтенда не придется переписывать код, если в ответах бэкенда что-нибудь поменяется.

Так же, имея актуальные схемы, можно генерировать документацию к API.

Тестирование API выполняется через Postman и написанием автотестов, в рамках проверки функциональности системы, используя язык программирования Python.

Логин/пароль к проекту в Postman: getpostman.com@gmail.com/Cpapsm2017 (JBEDBuKx)

Ссылка к тестам на gitlab: <https://gitlab.fix.ru/cpapsm/cpapsm-qa/tree/develop>

Исследовательское тестирование

Проводить время от времени всей командой, для поиска багов, неучтенных в регрессионном наборе.

Интеграционное тестирование

Результаты интеграционных тестов: <http://srv-00.infra.ivasp.de/jenkins/view/All/job/run-it-test/lastCompletedBuild/testReport/>

Тестирование отказоустойчивости

Проверка остановки, обновления и перезагрузки серверов под нагрузкой, без потери пользователей и подписок SWS/ SMPP servers



CPAPSM-1545 - Получение подробных данных проблемы...

СТАТУС

Задача на ручную проверку.

Приемочное тестирование

Во время планирования задач на спринт сразу же обговаривать приемочные критерии. QA команда, после создания задачи на разработку аналитиком, пишет и прикрепляет к задаче тест кейсы. По статусу данных тест кейсов QA команда определяет, брать задачу на тестирование или нет. Разработчик должен на DEV стенде успешно прогнать прикрепленные к задаче тест кейсы.

По возможности QA сразу автоматизирует новый функционал. Фича будет считаться завершенной, если пройдет приемочные тесты и набор регрессионных тестов.

Тестирование производительности, нагрузочное тестирование, стресс-тестирование

- определить метрики производительности совместно с аналитиками
- хранить результаты замеров производительности для каждого билда/релиза. Это даст возможность оперативно выяснить какое изменение в проекте стало причиной просадки производительности.
- перед релизом проводить стресс-тестирование. Сможем точно знать, какую нагрузку выдержит текущая конфигурация железа.
- тестирование стабильности. запускать в параллели много запросов в течение продолжительного периода времени. Сможем выяснить, нет ли утечек памяти и других проблем, связанных с длительной работой системы

Способ запуска Jmeter джоб на тестирование производительности - <http://srv-00.infra.ivasp.de/jenkins/view/All/job/run-jmeter-taurus/>

Сценарий, подающий нагрузку для регулярного автотеста редеплоя VASP под нагрузкой - <http://srv-00.infra.ivasp.de/jenkins/view/All/job/run-load-test-jmeter/>

Сценарий, подающий нагрузку на QA2 стенд - <http://srv-00.infra.ivasp.de/jenkins/view/All/job/run-load-test-on-qa2/>

Проверка статистики в service_counter по результатам тестирования редеплоя под нагрузкой - <http://srv-00.infra.ivasp.de/jenkins/view/All/job/check-load-test-statistic/>

Тестирование инфраструктуры

<http://wel.org.ua/it/testiruem-infrastrukturu>

- проверка синтаксиса конфигурации - <https://github.com/willthames/ansible-lint>
- проверять права на директории, была ошибка, связанная с этим

<https://habrahabr.ru/company/centosadmin/blog/303762/> - Ansible: тестируем плейбуки (часть 1)

<https://habrahabr.ru/company/centosadmin/blog/304814/> Ansible: тестируем плейбуки (часть 2)

Тестирование совместимости

Кроссбраузерное

Название приложения/события	Название браузера
CallCenter	Chrome
FrontOffice	IE
BackOffice	Firefox

	Safari
Подписка через LP с шаблоном	Chrome Mobile
	Chrome
	Android
	Yandex Browser
	Mobile Safari
	Opera Mini
	Opera
	IE
	IE Mobile
	Firefox
	Opera Mobile
	Mail.ru Chromium Browser
	Mobile Safari UIWebView
	Chrome Mobile iOS
	Facebook
	UC Browser
	Nokia Browser
	Firefox Mobile
	Dolfin
	Safari
	Edge
	Nokia Services (WAP) Browser
	Opera Coast
	Maxthon
	Iron
	BlackBerry

Открытие/редактирование отчетов

Название отчета	Название редактора
Детализированный отчет CPA	MS Excel
	Libre Office
Агрегированный GeneralReport	MS Excel
	Libre Office

Автоматизированное тестирование

Схема 1. Процесс автоматизации тестирования

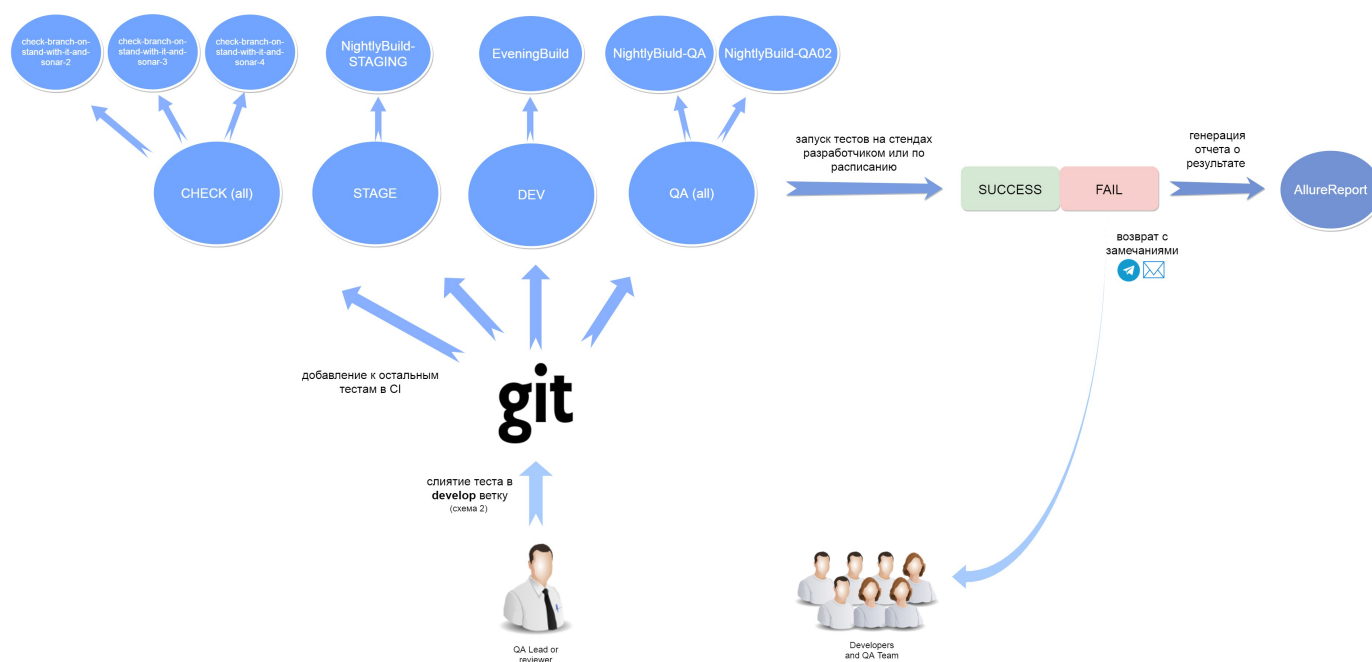
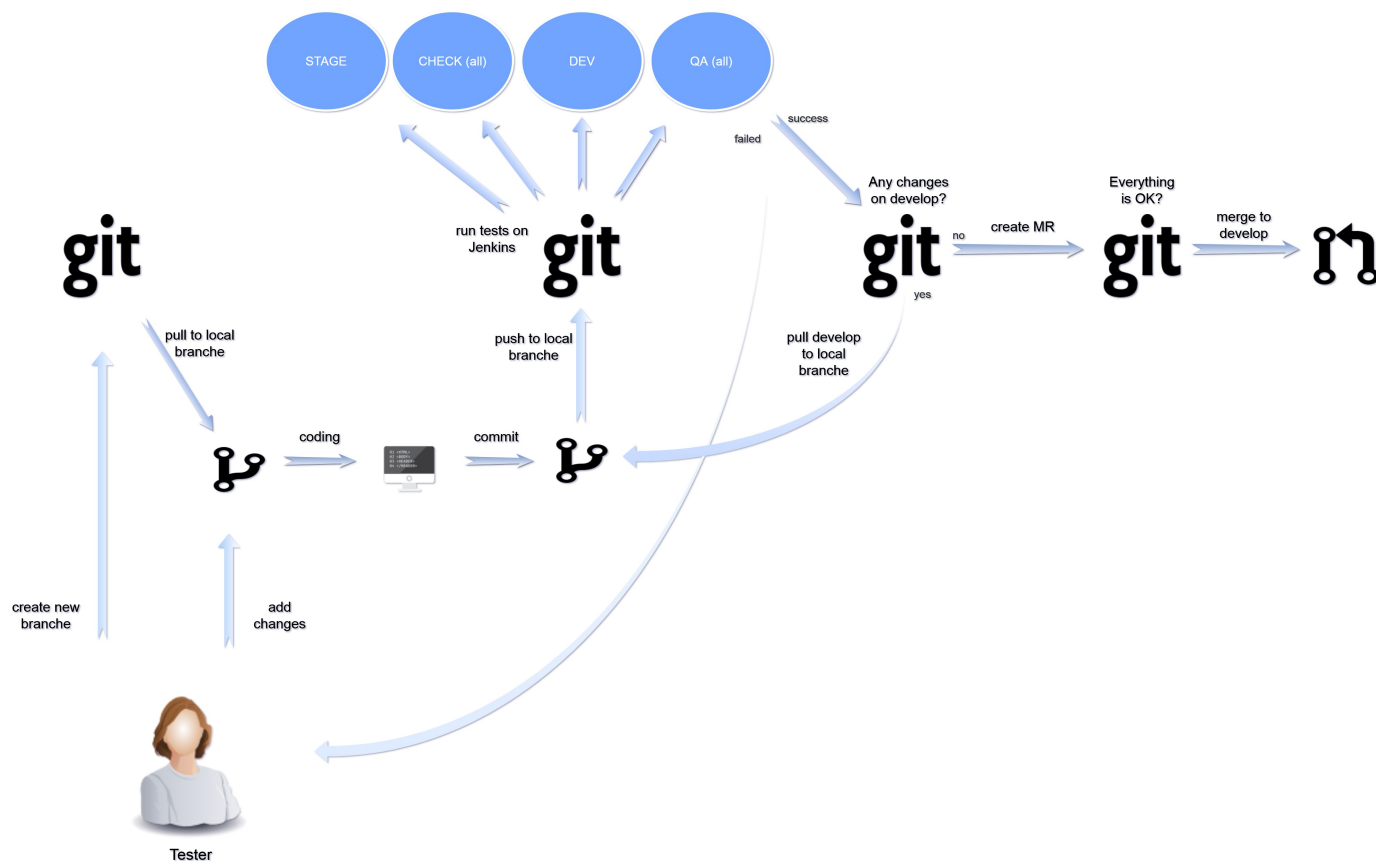


Схема 2. Детальная схема внесения новых изменений в develop ветку



Процесс автоматизации		
Этапы	Комментарии	Инструмент
Создание новой ветки из ветки develop или забрать изменения из develop в уже существующую ветку (Pull)		SourceTree, консоль
Написание автотеста		PyCharm
Отладка теста		PyCharm, AllureReport
Коммит готового теста в своей локальной ветке (Commit)		SourceTree, консоль
Отправка изменений в свою локальную ветку (Push)		SourceTree, консоль
Выполнение настроек для нового теста на каждом стенде: CHECK(all), DEV, QA(all), STAGE		БД (pgAdmin, DataGrid, aquaStudio), ZK-Web
Запуск теста в Jenkins CI на каждом стенде: CHECK(all), DEV, QA(all), STAGE		Jenkins
Если есть изменения в develop ветке, то выполнить слияние develop в локальную ветку		SourceTree, консоль
Создание запроса на слияние изменений (MR) в основную develop ветку		Gitlab
Проверка MR ведущим тестировщиком		Gitlab
Если нет замечаний, то слияние изменений в основную develop ветку		Gitlab
Если есть, то написание замечаний к MR и возврат к автотестеру		Gitlab

Инструменты		
Система контроля версий	Git	SourceTree
Система непрерывной интеграции	Jenkins CI	
Инструмент для статического анализа кода	SonarQube	
Инструмент для модульного тестирования	Kotlin	
Инструмент для функционального тестирования	Python	PyCharm
Инструмент для интеграционного тестирования	Kotlin	
Инструмент для тестирования производительности	Jmeter	
Инструмент для тестирования инфраструктуры	Ansible-lint	
Инструмент для тестирования интерфейса пользователя	Selenium WebDriverJs (Javascript)	
Инструмент для тестирования API	Postman	Postman
База данных	Postgresql	PGAdmin, AquaDataStudio, DataGrid
Просмотр логов	Kibana	http://logs.ivasp.de/app/kibana#/discover?_g=()&_a=(columns:!(_source),index:%27lo логин/пароль: vaspteam/Pamdijask
ZK конфиги	HUE	http://srv-05.infra.ivasp.de:8889/node?path=/cpapasm-qa02/qa02vaspdn01.ivasp.de:2181,qa02vaspdn02.ivasp.de:2181,qa02vaspdn03.ivasp.de:218

Проверка пользовательских сценариев перед демо

Проводить и проверять все критичные для бизнеса сценарии перед демо. Аналитики предоставят сценарии использования системы, которые критичны для бизнеса и команда QA перед демо должна подтвердить, что все они проходят без ошибок и конечные пользователи не получат ошибок.

Мониторинг

- Health check: BO, FO, CC - небольшие сценарии, запускать их и делать небольшие сценарии (логин, действие,

- логоут)брауз
- Zabbix
- Realtime данные по производительности, grafana
- мониторинг баз данных, выявлять тяжелые запросы в postgresql <https://habrahabr.ru/post/234481/> [https://habrahabr.ru/post/13992/](https://habrahabr.ru/company/okmeter/blog/311028/)

Деплой

Deploy тестового стенда: http://srv-00.infra.ivasp.de/jenkins/view/All/job/vasp_release_deploy/

- файл version.txt с версией компонентов и датой релиза. На данный момент версию можно посмотреть только в интерфейсе backOffice
- Post-deploy проверка: все сервисы/сервера обновились и запустились, проверка Tested задач на проверку работоспособности нового функционала
- автоматический health check

Чек-лист тестов перед релизом

- ☐ Прошли все приемочные тесты
- ☐ Билд прошел полный набор регрессионных тестов
- ☐ Проведено кроссбраузерное тестирование
- ☐ Нагрузочное/стресс тестирование

Полезные ссылки

Техническая часть	
Домены и адреса к разным приложениям VASP	STAND DOMAINS &URLs
Методы API	API URL
QA2	QA02
Тестовые сценарии	https://jira.fix.ru/secure/Tests.jspa#/
Организационные моменты	
	Рекомендации по результатам ревью 2016 (QA)