Основные обязанности на период моего отсутствия:

1. TCC-3DW
   1. Текущие задачи 3DW: Milestone + подзадачи + регрессионное тестирование для каждого Milestone. Выполняется на Iodine и др. серверах, собираемых разработчиками. Задача выполняется 1-2 тестировщиками.
      1. После того как пройдет тестирование сервер отправляется на review to Scott’s team.
      2. После review, обновляется код на beta stack и на load stack.
      3. После обновления load stack необходимо выполнить цикл нагрузочного тестирования и сравнить метрики.
   2. Load 3DW: Выполнение нагрузочного тестирования на load stack. Тестирование выполняется на Blazemeter. Скрипты тестирования разработаны. Возможно потребуется дополнить их. Лучше назначить кого-то одного на эту задачу – я предлагаю Оксану. Для выбранной конфигурации необходимо выполнить:
      1. Search тесты, на основе google stat.
      2. Search by tags.
      3. Search by q.
      4. Search by titles.
      5. Search & get actions.
      6. Get actions (models/collections/binaries)
      7. Set actions.
      8. Rendering actions.
      9. Search, based on the GData.
      10. Profile script, based on the use cases.

Дописать шаги.

* 1. Тестирование текущих задач ТСС. Здесь по приоритетам.
  2. Update тестовой документации. Концепт разрабатывает Илья Демченко. Он является координатором по этой задаче. По возможности выделите + 1-2 человек ему в помощь.
  3. Нагрузочное тестирование ТСС. Снятие метрик для релиза. Нужно будет поднять релизный стек и выполнить тестирование. Возможно потребуется доработать какие-то скрипты. Координатором в этой задаче является Валентин Гусев. Ему в помощь Галя + Даниил. Скрипты:
     1. По API запросам. Здесь контроль response time.
     2. Profile scripts.
  4. Quick data. Закончить скрипт, запустить, обновить. Координатор – Александр Малышкин.

1. DMD
   1. Здесь работаете вы и Оля.
2. Тестирование других проектов.
3. Оценка перспективных проектов. С этим обращается Аня Васенко или аналитики. Можете сами, а можете привлекать к оценке Валентина.
4. Оценка возникающих задач. Если задачи не глобальные, то отдаете на оценку тестировщикам проекта.
5. Запрос на распределение тестировщиков по проектам. Запрос от Антона. Согласовать с ним. На ТСС норма 200 тестерских часов в неделю.
6. Премиальные формы (разработчики до 08.11 и тестировщики по 09.11)
7. Контроль выработки. Еженедельно.
8. Контроль попадания в оценку. Просмотр задач в JIRA и redmine
9. Просмотр открытых дефектов.
10. Контроль работы Даниила (Hello ISSART + TCC).
11. Проведение собеседований.
12. Посещение руководительских совещаний.
13. Решение оперативных, возникающих вопросов: менеджеры, разработчики, HR-ы. Реже, аналитики, бухгалтерия, системные администраторы.

Тестирование Load Stack

1. 3dwload.releaseconnectedsite.com
2. Основная задача - <http://issues.apdbox.com/browse/TCCTWO-25490>
3. Выполняются скрипты с <http://www.blazemeter.com> . Login – [bquintrell@1apd.com](mailto:bquintrell@1apd.com) , password – cloud4apd
4. Скрипты перечислены в 1b. В Git путь – ./3DW/load testing/
5. Доступ к серверам с моей машины.
6. Доступ к консоли Amazon - <https://www.amazon.com/ap/signin?openid.assoc_handle=aws&openid.return_to=https%3A%2F%2Fportal.aws.amazon.com%2Fgp%2Faws%2Fdeveloper%2Fregistration%2Findex.html&openid.mode=checkid_setup&openid.ns=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0&openid.identity=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&openid.claimed_id=http%3A%2F%2Fspecs.openid.net%2Fauth%2F2.0%2Fidentifier_select&action=&disableCorpSignUp=&clientContext=&marketPlaceId=&poolName=596638996366&authCookies=&pageId=aws.iam&siteState=&accountStatusPolicy=P1&sso=&openid.pape.preferred_auth_policies=MultifactorPhysical&openid.pape.max_auth_age=3600&openid.ns.pape=http%3A%2F%2Fspecs.openid.net%2Fextensions%2Fpape%2F1.0&server=%2Fap%2Fsignin%3Fie%3DUTF8&accountPoolAlias=596638996366&forceMobileApp=0> , login: [kfirsanov@issart.com](mailto:kfirsanov@issart.com) , password: A#GxZ&&eel$#
7. Скрипты поиска базируются на статистике Google. Для каждого из скриптов нарезается определенный кусок файла статистики, так чтобы числа отрезанных записей хватило бы для выполнения 1 часового теста. Приложение для резки статистики: static\_parser-0.4.jar
8. Скрипты различаются по количеству одновременно подсоединенных пользователей.
9. Есть отдельные варианты скриптов для запросов с totalCounter=true, totalCounter=false и для запросов с сокращенным response.
10. Пример документа-отчета по нагрузочному тестированию в тикете задачи.

Премиальные формы разработчиков.

1. Форма скачивается с Google disk.
2. Используется классификатор премия-лишительных дефектов.
3. Дефекты выбираются из Redmine и JIRA (TCC, DMD).
   1. В Redmine:
      1. строится отчет за период по всем проектам, в которых участвовали тестировщики.
      2. По каждому из проектов выбираются созданные дефекты за период (Tracker: Bug & Defect)
      3. Оценивается каждый из дефектов проета. Дефект привязывается либо к feature или к задаче регрессионного тестирования.
      4. Если дефект привязывается к feature, то проверяется спецификация feature, и для каждого дефекта проверяется не было ли ситуации, когда дефект не был исправлен с первого раза.
      5. При попадании на премию, информируется разработчик, обсуждаются причины дефекта с ним.
      6. Если разработчик принимает, что работа была не качественной, то ему ставится в ведомости «Нет». В спорных ситуациях привлекать для разрешения проблемы менеджера проекта или старшего менеджера проекта.
      7. Если разработчик не создал дефектов, попадающих под классификатор, то ему в ведомости ставится «Да».
   2. JIRA
      1. Просматриваются все задачи, с которыми работали тестировщики за отчетный период.
      2. В features просматриваются все открытые дефекты. Мы их пишем пока в комментарии. Выбираются только те комменты-дефекты, которые были открыты за отчетный период.
      3. Сравниваются с description (де факто требованиями). Проверяется, что при ре-тестинге дефект был исправлен.
      4. Проверяются баги, на предмет того, что они были исправлены с первого раза.
      5. При попадании на премию, информируется разработчик, обсуждаются причины дефекта с ним.
      6. Если разработчик принимает, что работа была не качественной, то ему ставится в ведомости «Нет». В спорных ситуациях привлекать для разрешения проблемы менеджера проекта или старшего менеджера проекта.
      7. Если разработчик не создал дефектов, попадающих под классификатор, то ему в ведомости ставится «Да».
4. Для тех разработчиков, которые работают на проекта, где нет тестирования, в форме не ставится ничего.

Премиальные формы тестировщиков.

1. Форма у вас есть, вам отправляли ее HR в октябре.
2. По каждому из проекту из Redmine, на которых работали тестировщики, выбираются только задачи с tracker Defect. Если таковых нет, то все ОК. Если нет, то идем к менеджеру проекта и узнаем, появились ли эти дефекты, как результат некачественной работы тестировщика или есть какая-то иная причина. Для TCC оцениваются только результаты выкатки: открыли ли заказчики дефекты или нет.
3. Строится отчет по всем задачам, решенным тестировщиком за отчетный период.
4. Выбираются только закрытые задачи.
5. Фиксируются estimation time и spent time. Вычисляется усредненное время.