

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»
Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.04.03 Прикладная информатика
(код и наименование направления подготовки)

Управление корпоративными информационными процессами
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему Разработка и оптимизация дашбордов в процессе приемочного тестирования
на основе опыта в IT-компаниях

Обучающийся

О.В. Дунаевская

(Инициалы Фамилия)

(личная подпись)

Научный
руководитель

канд. техн. наук, доцент О.В. Аникина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

Оглавление

Введение.....	3
Глава 1 Теоретическая и методологическая база организации приёмочного (интеграционного) тестирования.....	7
1.1 Особенности приемочного (интеграционного) тестирования	7
1.2 Понятие дашборда и его роль в IT-компаниях	20
1.3 Обзор существующих методов оптимизации дашбордов	34
Глава 2 Анализ опыта работы с дашбордами при приемочном тестировании в IT-компаниях	43
2.1 Анализ деятельности компании	43
2.2 Использование дашбордов в процессе приемочного тестирования.....	49
2.3 Анализ проблем, выявленных в опыте работы.....	56
Глава 3 Разработка и оптимизация дашбордов для процесса приемочного тестирования.....	62
3.1 Проектирование дашбордов, учитывающих специфику процесса приемочного тестирования	62
3.2 Оценка эффективности разработанных дашбордов	72
Заключение	79
Список используемых источников.....	81
Приложение А Вопросы для анкетирования инженеров по проверке качества ПО	87
Приложение Б Бизнес-план по оптимизации процесса приёмочного (интеграционного) тестирования.....	88
Приложение В Преимущества и недостатки «дашборда для руководителей» и «дашборда для всех участников рабочего процесса»	90
Приложение Г Уровни тестирования по степени детализации приложения.....	93

Введение

Информационное общество ставит перед IT-компаниями все более сложные и высокие требования к качеству разрабатываемых программных продуктов. Крупные компании с целью формирования положительного имиджа и улучшения репутации заинтересованы поставлять конечному пользователю готовые и удобные приложения и сервисы. Поэтому для выявления дефектов, допущенных в процессе разработки, нанимают инженеров по тестированию и проверке качества программного обеспечения. В связи с этим, тестирование занимает важное место в жизненном цикле разработки программных продуктов. Особенно следует выделить приёмочное (интеграционное) тестирование. Его результаты являются решающими в принятии решения о выдаче продукта или его нового релиза пользователям.

IT-компании заинтересованы постоянно повышать эффективность и качество приемочного тестирования, в связи с этим ведутся разнонаправленные работы: от выбора методологии рабочего процесса и повышения квалификации инженеров до обеспечения рабочего пространства материально-техническими средствами.

При управлении командой руководитель встречается с ситуацией, когда не имеет достаточного количества времени на ежедневное распределение рабочих задач и отслеживание статуса их выполнения для каждого сотрудника в отдельности [20]. Поэтому так важно прибегать к помощи современных инструментов мониторинга и выводить рабочую команду на уровень самоорганизации. Важным аспектом в этом процессе является наличие надежных и информативных средств для отслеживания и анализа результатов тестирования, например, таких, как дашборды (панели мониторинга).

Актуальность исследования данной темы обусловлена быстрым развитием технологий и появлением всё большего количества инструментов для разработки дашбордов, применение которых предоставляет отличную

возможность для улучшения процесса тестирования, оптимизации рабочих активностей и повышения надежности программного обеспечения.

Объектом исследования является процесс приёмочного (интеграционного) тестирования.

В качестве предмета исследования мы рассматриваем разработку и оптимизацию модели дашборда для повышения эффективности приёмочного (интеграционного) тестирования.

Целью данного исследования является создание модели дашборда, содержащего актуальную и максимально концентрированную информацию о процессе приёмки, интересную всем участникам рабочего процесса и управления.

Данная цель определила необходимость решения следующих задач:

- исследовать доступные теоретические материалы об организации мониторинга в процессе тестирования ПО;
- создать модель дашборда, отражающую все аспекты приёмочного процесса в актуальном состоянии;
- подобрать необходимые метрики для мониторинга приёмочного (интеграционного) тестирования;
- описать фильтры на основе выбранных метрик;
- апробировать данную модель в периоды прохождения производственной практики и научно-исследовательской работы;
- обучить рабочую группу по тестированию программного обеспечения применять модель для принятия решений о приоритизации работ;
- сделать вывод об эффективности предложенной модели дашборда;
- распространить использование подобной модели на аналогичные процессы внутри компании ООО «НетКрэкер».

Нами была сформулирована следующая гипотеза: Применение дашбордов, отражающих необходимую и актуальную информацию для исполнителей тестирования, позволяет вывести группу проверки качества

принимаемого программного обеспечения на уровень самоорганизации в части получения рабочих задач и их приоритизации, улучшает целеполагание и повышает ориентирование в рабочем процессе.

В данной работе используются следующие методы исследования: обобщение справочной и научной литературы, теоретический анализ, классификация, систематизация, сравнительный и статистический анализ, объектно-ориентированный подход к моделированию систем управления.

Чаще всего в бизнесе предпочитают организацию дашбордов «снизу-вверх», т.е. информация передаётся от руководителя команды к менеджерам проекта и выше. Не рассматривается возможность использования дашборда непосредственными исполнителями работ. Мы исследовали варианты применения дашбордов для самоорганизации рабочей команды на примере приёмочных работ по проверке качества программного обеспечения в итеративной методологии. Описание примеров для выстраивания модели дашборда таким образом, чтобы информация была доступна на всех уровнях исполнения работ и управления, определяет новизну исследования.

Практическая значимость данного исследования состоит в возможности применения разработанной модели дашбордов в рамках приёмочного тестирования внутри компаний, разрабатывающих программное обеспечение.

Работы в области организации дашбордов для ведения бизнеса, опубликованные зарубежными и отечественными авторами, такими как Т. Hamilton, А. Sarikaya, М. Correll, J. Mackinlay, Голованова О.С., Крупина В.В., Куликов С.С. и другими, а также техническая и нормативная документация об организации тестирования составляют методологическую базу исследования.

Данное исследование охватывает период с 2022 по 2023 годы. Оно проводилось в три этапа:

На начальном этапе была определена тема исследования, выполнен сбор и анализ информации, поставлена цель и задачи, определены предмет, объект и гипотеза исследования.

На следующем этапе проведён анализ существующих дашбордов, разработана первичная модель дашборда, обучена рабочая группы по проверке качества программного обеспечения, а также написана и опубликована научная статья по теме исследования.

На последнем этапе выявлены ограничения рабочей среды и предложена оптимизация разработанной модели.

На защиту выносятся:

- разработанная модель дашборда,
- анализ эффективности разработанной модели.

В контексте данной темы исследования опубликована 1 статья «Дашборд как основа самоорганизации команды во время приёмочных работ по проверке качества программного обеспечения» [9], результаты исследования были представлены на IX Международной научно-практической конференции (школы-семинара) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук».

Работа состоит из введения, трех глав, заключения, изложенных на 80 страницах, списка используемой литературы (48 наименований) и 4 приложений.

Глава 1 Теоретическая и методологическая база организации приёмочного (интеграционного) тестирования

1.1 Особенности приемочного (интеграционного) тестирования

Под тестированием понимается «процесс анализа программного продукта и сопутствующей документации с целью выявления дефектов и повышения качества программного обеспечения» [16].

В соответствии с IEEE Std 829-1983, тестирование – «это процесс анализа программного обеспечения, направленный на выявление отличий между его реально существующими и требуемыми свойствами (дефект) и на оценку свойств программного обеспечения» [2, 132].

Различают следующие уровни тестирования по степени детализации приложения [33]:

- компонентное (модульное) тестирование,
- интеграционное тестирование,
- системное тестирование.

Данные уровни тестирования описаны в Приложении Г.

Типы тестирования [33]:

- функциональное (осуществляется на всех уровнях тестирования; тесты включают в себя проверки по оценке функций программного обеспечения и поведению системы) [33, 41];
- нефункциональное (проверка работоспособности программного обеспечения или системы; выполняется «для оценки таких характеристик системы и программного обеспечения, как удобство использования, производительность или безопасность» [33, 42]; также должно выполняться на всех уровнях тестирования) [35];
- тестирование методом белого ящика («основывается на внутренней структуре системы или ее реализации» [33, 42]; предполагается, что тестировщик имеет доступ к просмотру программного кода);

– тестирование методом чёрного ящика (основывается на взаимодействии с интерфейсом системы, доступном конечному пользователю; тестировщик имеет только доступ к тому интерфейсу системы, с которым работают конечные пользователи, либо к интерфейсу программы, работающей непосредственно с тестируемым программным обеспечением) [34];

– тестирование, связанное с изменениями (различают подтверждающее тестирование, направленное на проверку работоспособности функционала после исправления дефекта, и регрессионное, направленное на проверку работоспособности всего компонента, в который вносились правки, и зависимых от него компонент).

Выполнение любого из вышеописанных типов тестирования возможно выполнить на любом уровне тестирования [36].

Современное программное обеспечение, разрабатываемое ИТ-компаниями, состоит из многих компонент, очень различающихся по функционалу, но связанных между собой. Таким образом, каждый новый релиз требует тщательного тестирования, и поэтому на завершающих этапах работ наиболее востребованным является уровень интеграционного тестирования [21].

Согласно, «Стандартному глоссарию терминов, используемых в тестировании программного обеспечения» [38] по материалам ISTQB (International Software Testing Qualifications Board):

Интеграционное тестирование (integration testing): «тестирование, выполняемое для обнаружения дефектов в интерфейсах и во взаимодействии между интегрированными компонентами или системами» [19]. Данный формат предполагает, что все компоненты, связанные с определенным узлом, являются основой для интеграционного тестирования.

Процесс объединения компонентов или систем в более крупную структуру называется интеграцией [37]. Целью тестирования интеграции компонентов является выявление дефектов в интерфейсах и между интегрированными компонентами, т.е. важно определить, что обновлённые

компоненты позволяют всему ПО работать качественно и корректно, в том числе при взаимодействии с внешними системами (интернет и т.д.).

Объект интеграционного тестирования – это компонент или система, подвергаемая тестированию в рамках процесса интеграционного тестирования. Объект интеграционного тестирования может быть программным модулем, библиотекой, службой, веб-приложением или даже целым информационной системой. Основная задача интеграционного тестирования – проверить взаимодействие различных компонентов или модулей программного обеспечения [1].

Системное интеграционное тестирование определяется как тип тестирования программного обеспечения, проводимый в интегрированной аппаратной и программной среде для проверки поведения всей системы. Это тестирование полной интегрированной системы для оценки ее соответствия установленным требованиям [5].

В рамках тестирования интегрированных компонентов, модулей или систем чаще всего проверяются:

- интеграция компонентов: Основная цель интеграционного тестирования – проверить взаимодействие и корректную работу между связанными компонентами программного обеспечения.
- последовательность тестирования: Тестирование проводится в определенной последовательности, в которой компоненты интегрируются и тестируются на соответствие ожидаемому поведению.
- тестирование интерфейсов: Одной из основных задач интеграционного тестирования является проверка правильности работы интерфейсов между компонентами.
- обработка данных: Проверка правильности передачи данных между компонентами и их корректности в процессе взаимодействия.
- проверка функциональности: Интеграционное тестирование включает проверку соответствия функциональных требований и ожидаемого поведения при взаимодействии компонентов.

- обработка ошибок и исключений: Интеграционное тестирование включает проверку обработки ошибок и исключительных ситуаций в процессе взаимодействия компонентов.
- тестирование конфигурации: Проверка правильности работы при различных конфигурациях системы.
- тестирование производительности: Оценка производительности системы при интеграции компонентов и взаимодействии между ними.
- тестирование безопасности: Проверка безопасности системы при взаимодействии компонентов.
- тестирование совместимости: Проверка совместимости различных компонентов программного обеспечения.
- автоматизация тестирования: Интеграционное тестирование может быть автоматизировано для повышения эффективности и точности тестирования.

Перечислим типы интеграционного тестирования.

Подход «большого взрыва» («Big Bang Approach» [47]) – это подход к интеграционному тестированию, при котором все и компоненты интегрируются в одно время, а затем проверяются как единое целое. Процесс интеграции не будет стартован, пока все компоненты в блоке не будут завершены.

Поэтапный подход («Incremental Approach» [47]) осуществляет тестирование путем интеграции двух или более компонент, которые связаны друг с другом логически, а затем проверяет приложение на правильность функционирования. После этого постепенно подключают другие зависимые модули, и процесс выполняется до тех пор, пока все логически связанные модули не будут интегрированы и успешно протестированы. Здесь уместно использовать заглушки и драйверы – фиктивные программы, используемые для упрощения проверки программного обеспечения. Эти программы замещают отсутствующие компоненты или модели. Они не реализуют всю

логику программирования программного модуля, но имитируют обмен данными с вызывающим модулем во время тестирования [6].

Поэтапный подход разделяют на подтипы:

– Нисходящий, подход «сверху вниз» («Top Down Approach» [47]) – стратегия, в которой тестирование проводится, следуя потоку управления программной системой. Сначала тестируются модули высоких уровней, а затем интегрируются модули уровнями ниже для проверки функциональности программного обеспечения. Заглушки используются для тестирования, если некоторые модули не готовы.

– Восходящий, подход «снизу вверх» («Bottom Up Approach» [47]) – метод, в котором сначала в проверку включаются модули нижних уровней. Эти модули используются для тестирования вышестоящих компонент и максимально точно моделируют бизнес заказчика. После того как модули нижнего уровня интегрированы и проверены, для вышестоящего уровня формируется следующий этап тестирования. Процесс продолжается до тех пор, пока в интеграцию не будут включены и проверены все модули и компоненты верхнего уровня.

– Сэндвич-подход («Sandwich Approach» [47], «Hybrid Integration Testing») – метод, в котором модули верхнего и нижнего уровня интегрируются и тестируются одновременно, как система. Это сочетает нисходящий и восходящий подходы, поэтому называется гибридным. Данный подтип тестирования использует как заглушки, так и драйверы.

Конкретные особенности интеграционного тестирования могут изменяться или дополняться в зависимости от проекта, требований к системе и используемых технологий [7].

Ниже приведена таблица преимуществ и недостатков для каждого типа интеграционного тестирования (таблица 1).

Таблица 1 – Преимущества и недостатки типов интеграционного тестирования

Тип интеграционного тестирования	Преимущества	Недостатки
Подход «большого взрыва»	Удобен для небольших систем	<p>Локализация неисправности затруднена. Количество интерфейсов, которые необходимо протестировать, огромно, и при таком подходе можно легко пропустить часть тестируемых интерфейсов.</p> <p>Группе тестирования будет выделено меньше времени для выполнения проверок, поскольку интеграционное тестирование допускается только после того, как все компоненты и модули будут разработаны.</p> <p>Одновременная проверка всех модулей не принимает во внимание необходимость изолирования и тестирования в приоритетном порядке критических и периферийных модулей.</p>
Поэтапный: нисходящий подход	<p>Локализация неисправности проще. Возможно получение раннего прототипа. Критические модули проверяются в первую очередь; в приоритетном порядке можно найти и исправить ключевые недостатки функционала.</p>	<p>Требуется много заглушек. Модули более низкого уровня тестируются некорректно.</p>
Поэтапный: восходящий подход	<p>Проще локализация неисправности. Время не затрачивается на ожидание завершения разработки всех модулей.</p>	<p>Модули верхнего уровня архитектуры (критические), контролирующие поток приложений, тестируются в последнюю очередь и подвержены критическим дефектам.</p> <p>Получение раннего прототипа невозможно.</p>

Функционал программного обеспечения зависит от конфигурации, запрошенной заказчиком. Конфигурация выстраивается из набора компонент программного обеспечения. Каждая компонента создаётся в отдельной команде, в которую входят проектные менеджеры, бизнес-аналитики,

разработчики и инженеры по контролю качества, а затем выдаётся для приёма в релиз [48].

Приемочное (интеграционное) тестирование выполняется для проверки работоспособности интеграций модулей системы, функциональности, производительности программного обеспечения, его безопасности и совместимости с другими компонентами. В процессе приемочного тестирования инженеры убеждаются в том, что разработанный продукт соответствует требованиям заказчика, а также оптимизируют программное обеспечение или его компоненты для повышения удобства их использования и эффективности. Основная цель приемочного тестирования заключается в подтверждении качества программного обеспечения в целом, качества принимаемых компонент, включённых в конфигурацию релиза, и обновлений к ним, а также поддержании регрессионного списка тестовых сценариев в актуальном состоянии.

Критерии начала интеграционного тестирования могут различаться в зависимости от проекта и его требований. Ниже перечислены общие критерии, которые могут быть использованы для определения начала интеграционного тестирования:

- окончание модульного тестирования: Тестирование интеграции следует начинать после успешно завершённого модульного тестирования компонентов системы. Это гарантирует, что каждый модуль функционирует корректно и готов для интеграции.

- определение последовательности интеграции: Необходимо определить порядок и способ интеграции компонентов системы. Это позволяет планировать и выполнять интеграцию поэтапно и контролировать прогресс тестирования.

- наличие готовой интеграционной среды: Для успешного проведения интеграционного тестирования необходимо иметь готовую интеграционную среду, которая включает все необходимые компоненты и настройки.

– определение целей и ожидаемых результатов: Перед началом интеграционного тестирования необходимо определить цели и ожидаемые результаты. Это поможет оценить процесс интеграции и успешность тестирования.

– подготовка тестовых случаев: Необходимо разработать тестовые случаи, которые позволят проверить взаимодействие между интегрируемыми компонентами. Тестовые случаи должны быть покрывающими и охватывать все функциональные требования.

– готовность тестовой команды: Тестовая команда должна быть готова к проведению интеграционного тестирования. Это включает наличие необходимых навыков и ресурсов, а также понимание особенностей интеграционного тестирования.

– утверждение руководства проекта: Перед началом интеграционного тестирования, руководство проекта должно утвердить план тестирования, включая последовательность интеграции, цели и ожидаемые результаты.

– завершение предварительных проверок: Перед началом интеграционного тестирования необходимо завершить все предварительные проверки, например, проверку конфигурации и наличие необходимого оборудования.

Учет этих критериев поможет обеспечить успешное начало интеграционного тестирования и эффективное выполнение этого этапа разработки проекта.

Рассмотрим основные активности процесса выполнения приёмочного (интеграционного) тестирования.

Планирование тестирования определяет «цели тестирования и подход к достижению целей тестирования с ограничениями, налагаемыми контекстом (например, определение подходящих методов тестирования и задач, а также формирование графика тестирования для соблюдения крайнего срока). Планы

тестирования могут быть пересмотрены на основе обратной связи от мониторинга и контроля» [32, 19].

Мониторинг тестирования предусматривает сравнение фактического хода работы с планом тестирования, при использовании метрик, заявленных к отслеживанию. «Контроль тестирования подразумевает принятие мер, необходимых для достижения целей плана тестирования (который может быть обновлен с течением времени)» [32, 19]. Мониторинг и контроль тестирования поддерживаются оценкой критериев выхода, которые в некоторых жизненных циклах называются критериями готовности [32]. Составляются отчёты о ходе тестирования, в которых заинтересованным лицам раскрывается информация о прогрессе тестирования по сравнению с планом, включая отклонения от плана.

В процессе анализа тестирования определяется базис тестирования функций и устанавливаются соответствующие тестовые условия. Данный анализ определяет измеримые критерии покрытия тестовых сценариев и устанавливает, что именно будет протестировано.

Для уменьшения количества пропущенных важных проверок и определения точных тестовых условий полезно применять такие методы тестирования как тестирование на основе черного ящика, белого ящика и на основе опыта инженера по проверке качества программного обеспечения. Важным потенциальным преимуществом в поддержании качества программного обеспечения является «раннее выявление дефектов в ходе анализа тестирования» [33].

При проектировании тестов тестовые условия преобразуются в наборы высокоуровневых тестовых сценариев и другие тестовые инструменты. Проектирование тестов определяет, как именно будут протестированы проверки. Как и в случае с анализом тестирования, проектирование тестов может выявить сходные типы дефектов в базисе тестирования. Выявление дефектов при проектировании тестов также является важным потенциальным преимуществом в поддержании качества программного обеспечения.

Активности по реализации тестов направлены на создание и/или подготовку необходимого тестового обеспечения для выполнения проверок, включая приоритизацию и последовательность выполнения тестовых сценариев. Во время этой активности сценарии и процедуры выстраиваются в определенном порядке, чтобы облегчить проведение тестирования.

Задачи по реализации и проектированию тестов могут быть объединены – это допустимая практика.

В процессе выполнения тестов, наборы сценариев запускаются в соответствии с установленным расписанием. Мануальное тестирование выполняется в течение рабочего времени инженеров по проверке качества программного обеспечения, автотесты могут выполняться круглосуточно.

При завершении тестирования собирают данные о выполненных работах с целью обобщить опыт всех членов рабочей группы, выявления закономерностей и исключений тестового обеспечения или сбора иной соответствующей информации. «Активности по завершению тестирования происходят на вехах проекта, например, при завершении выпуска релиза программного обеспечения системы, завершении (отмене) проекта тестирования, окончании итерации проекта с гибкой методологией (например, как часть итогового совещания), завершении уровня тестирования, или завершении сопровождения релиза» [33, с. 23].

Критерии, описывающие завершения интеграционного тестирования, могут различаться в зависимости от проекта и его требований. Приведём восемь общих критериев, которые могут использоваться для определения завершения интеграционного тестирования:

- Завершение всех запланированных интеграционных тестов: Все запланированные тесты должны быть выполнены и пройдены успешно. Это включает тестирование всех необходимых сценариев и функциональных требований.

- Устранение выявленных дефектов: Все выявленные дефекты должны быть зарегистрированы, отслежены и исправлены. Интеграционное

тестирование должно быть завершено после устранения всех критических и высокоприоритетных дефектов.

- Успешное взаимодействие между компонентами: Взаимодействие между интегрируемыми компонентами программного обеспечения должно быть проверено и соответствовать ожидаемому поведению. Все интерфейсы должны работать корректно и передавать данные без ошибок.

- Покрытие функциональности: Интеграционное тестирование должно охватывать все функциональные требования, учитывая основные сценарии использования и возможные вариации.

- Успешное прохождение производительных тестов: Если в проекте требуется проверка производительности системы, необходимо выполнить и пройти все производительные тесты, установленные для интеграционного тестирования.

- Успешное прохождение тестов совместимости: Если в проекте имеются требования к совместимости системы с другими компонентами или окружением, необходимо выполнить и пройти все тесты, связанные с совместимостью.

- Согласованность с критериями приемки: Если в проекте предусмотрены критерии приемки, интеграционное тестирование должно соответствовать этим критериям. Успешное прохождение интеграционного тестирования может быть одним из условий для принятия системы.

- Утверждение руководства проекта: Перед завершением интеграционного тестирования, руководство проекта должно подтвердить, что все необходимые тесты выполнены, выявленные дефекты исправлены и система готова для продолжения следующего этапа разработки или релиза.

Эти критерии могут быть адаптированы в соответствии с требованиями и особенностями каждого конкретного проекта.

Метрики качества интеграционного тестирования помогают измерить эффективность и полноту тестирования процесса интеграции компонентов системы. Приведём несколько примеров таких метрик:

– **Покрытие интеграции:** Эта метрика измеряет процент компонентов системы, которые были протестированы в процессе интеграционного тестирования. Чем выше покрытие, тем больше компонентов было проверено.

– **Количество и природа обнаруженных дефектов:** Эта метрика отражает количество и типы дефектов, обнаруженных в процессе интеграционного тестирования. Это позволяет оценить степень проблем и ошибок, связанных с взаимодействием компонентов.

– **Стабильность системы:** Метрика стабильности измеряет способность системы функционировать без сбоев и сбоев после интеграции компонентов. Это может быть выражено в виде времени безотказной работы или количества сбоев.

– **Процент успешных сценариев интеграции:** Эта метрика отражает процент успешно пройденных сценариев интеграции. Успешное прохождение сценариев свидетельствует о правильной работе компонентов вместе.

– **Производительность:** Метрика производительности измеряет производительность системы после интеграции компонентов. Это может включать измерение времени отклика или пропускной способности системы.

– **Объем тестирования:** Эта метрика отражает количество тестовых случаев, применяемых в интеграционном тестировании. Большой объем тестирования может гарантировать более полное покрытие системы.

– **Время и ресурсы:** Метрика времени и ресурсов отражает затраты на проведение интеграционного тестирования, включая время, участие тестировщиков и требуемое оборудование.

– **Возврат инвестиций:** Эта метрика оценивает возврат инвестиций в проведение интеграционного тестирования. Она может быть связана с улучшением качества продукта, снижением рисков и повышением доверия к системе.

Эти метрики помогают оценить качество и эффективность интеграционного тестирования, а также выявить области для улучшения процесса интеграции компонентов системы.

Приемочное (интеграционное) тестирование играет ключевую роль в успешной разработке и внедрении системы. Этот процесс обеспечивает проверку работы компонентов и их взаимодействия, чтобы убедиться в работоспособности системы в целом.

Выявляются и исправляются ошибки, связанные с интеграцией компонентов системы (модулей, подсистем или даже взаимодействие с другими внешними системами). Каждый компонент может быть протестирован отдельно, но при интеграции между ними могут возникнуть проблемы, такие как несовместимость интерфейсов, неправильное взаимодействие или конфликты ресурсов. Приемочное (интеграционное) тестирование помогает выявить такие проблемы на ранних этапах разработки, что позволяет своевременно их исправить, и таким образом снизить риски и затраты в долгосрочной перспективе.

Происходит проверка корректности функционирования системы в целом. В процессе интеграции компонентов могут возникать не только технические проблемы, но и проблемы с бизнес-логикой или требованиями заказчика. Приемочное тестирование позволяет убедиться, что система соответствует ожиданиям заказчика и выполняет все функции, необходимые для успешной работы.

Приемочное тестирование способствует повышению качества системы. Проверка работы компонентов в реальных условиях интеграции позволяет выявить и исправить проблемы, которые могут быть непросто обнаружить на уровне отдельных компонентов. Это важно для создания стабильной, надежной и эффективной системы, которая будет работать без сбоев и проблем в реальной эксплуатации.

Кроме того, приемочное тестирование способствует повышению доверия к системе. Заказчики и пользователи должны быть уверены, что

система будет работать надежно и безопасно, их данные будут защищены, а функционал будет доступен в любых ситуациях. Приемочное (интеграционное) тестирование демонстрирует, что система проверена и готова к использованию, что увеличивает доверие к ее качеству и надежности.

Наконец, приемочное тестирование помогает снизить риски и избежать потерь, связанных с некорректной работой системы. Выявление и устранение проблем на ранних стадиях разработки существенно снижает возможность возникновения серьезных ошибок или сбоев в работе системы после внедрения. Это позволяет избежать потери времени, ресурсов и доверия со стороны пользователей [14].

В заключение, приемочное (интеграционное) тестирование играет важную роль в успешной разработке и внедрении системы. Оно помогает выявить и исправить проблемы, связанные с интеграцией компонентов, проверить корректность функционирования системы, повысить ее качество и надежность, увеличить доверие к системе и снизить риски. Поэтому, при разработке и внедрении системы, важной частью процесса является приемочное тестирование, которому следует уделить должное внимание и ресурсы.

1.2 Понятие дашборда и его роль в IT-компаниях

Составим представление о рабочих командах, взаимодействующих в процессе интеграционного тестирования. В случае, если программное обеспечение разрабатывается, тестируется и принимается внутри одной организации, данную работу непосредственно включены инженеры по проверке качества программного обеспечения, инженеры по поддержке тестового окружения, бизнес-аналитики, разработчики программного обеспечения, руководители рабочих групп (team leads) и продуктовые менеджеры [8]. Если же разработки ведутся для внешнего заказчика, то список рабочих групп может быть значительно расширен и включить менеджеров со

стороны заказчика, разработчиков других компаний, разрабатывающих компоненты, подлежащие интеграции, группу лиц из организации, предлагающей технические и технологические услуги приёмочного тестирования и других [9].

В рамках приёмочных работ один тестовый сценарий проходит множество этапов контроля у разных сотрудников (см. рисунок 1).

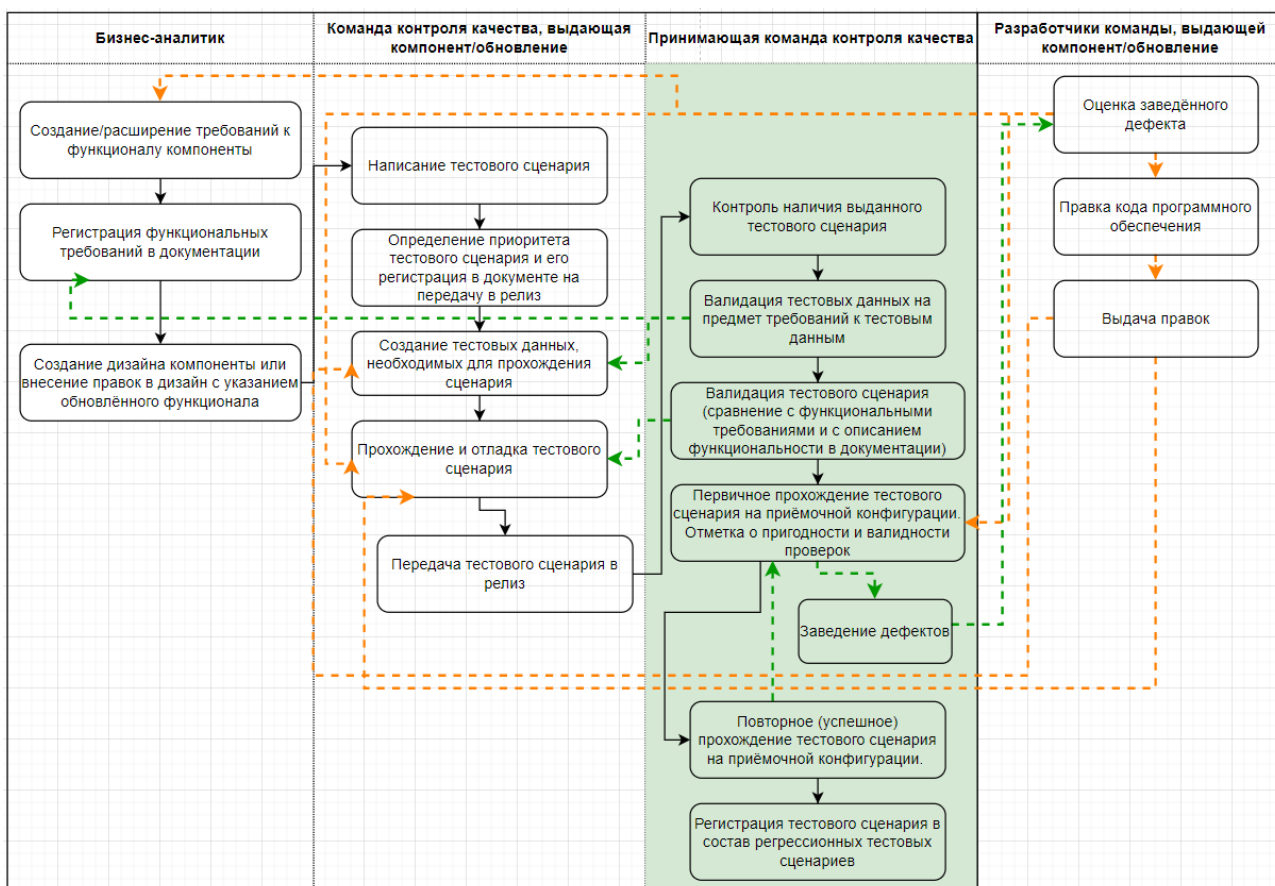


Рисунок 1 – Схема «Путь одного тестового сценария»

Сплошной линией обозначены основные действия. Зелёным пунктиром обозначены возможные действия, которые будут инициированы принимающей командой; оранжевым пунктиром обозначены возможные действия, которые будут инициированы командой, выдающей компонент или обновления к нему, – такие действия необходимо будет выполнить в случае

сбоя в тестовых данных, неточности/недостаточности тестового сценария или при выявлении дефекта.

В основе разработки тестового сценария лежат функциональные требования и дизайн-документация, которая предоставляется бизнес-аналитиками. Далее на этой основе команда инженеров по контролю качества, выдающая компонент или обновления к нему, формирует ручной или автоматический тестовый сценарий, создаёт необходимые данные для проведения проверки, описывает настройку конфигурации тестового окружения, определяет приоритет сценария и производит его отладку. Затем тестовый сценарий передаётся принимающей команде, т.е. в приёмочное тестирование [18].

Принимающая команда производит функциональные и интеграционные проверки на основе тестового сценария, проверяет соответствие требованиям, описанным в дизайн-документации, убеждается в том, что тестовый сценарий возможно пройти на рекомендованной конфигурации, и что при прохождении тестового сценария не встречены дефекты и иные нарушения. Если возникает необходимость правок тестового сценария, работа может быть возвращена команде инженеров по контролю качества, выдающей компонент или обновления к нему. Если будет найден дефект, то об этом сообщается в команду разработчиков, выдавшую компонент или обновления к нему. Любые правки проверяются повторно на предмет внесённых изменений и работоспособности тестового сценария.

Работа по приёмке тестового сценария завершается тем, что принимающая команда объявляет данный сценарий пригодным, а его проверки валидными, в соответствии с ранее заявленными целями тестирования.

Таких тестовых сценариев в рамках приёмочных работ может быть большое количество, также существует вероятность, что на один тестовый сценарий будет заведено более одного дефекта, что поднимает вопрос о проблемах распределения задач внутри команд, демонстрации актуального

состояния работ и отслеживании ключевых рабочих моментов. Данную проблему позволяют решить инструменты, предоставляемые JIRA (например, TMS – Test Management System for Jira), которые позволяют выгружать задачи разных типов и формировать с ними дашборды и отчёты [27].

Дашборд (dashboard, иначе – панель мониторинга, информационная панель) в переводе с английского – «приборная панель» Исторически название произошло от приборной панели автомобиля [15].

Использование дашбордов в бизнесе началось примерно в середине 1990-х годов. В этот период компании стали осознавать необходимость систематизации, визуализации и анализа своих данных для принятия обоснованных решений. «Раньше работал подход Highest Paid Person's Opinion – принятие решений на основе мнения руководителя, которое часто было субъективным. В противовес ему сформулировали подход Data Driven Decision Making (DDDM) – при нём решения принимают на основе данных» [22]. Использование этого подхода предполагает необходимость сбора данных и представления их в письменной форме. Это спровоцировало появление большого количества отчётов.

Для удобства работы с данными специалисты добавляли в отчёты различные визуализации. Таким образом появились дашборды. В настоящее время они применяются в многочисленных системах обработки данных, и продвинутые организации разрабатывают нужные им информационные панели под свои задачи [11].

Под дашбордом мы будем понимать инструмент, который не только визуализирует, но и анализирует рабочие данные. Дашборд позволяет лаконично описать большое количество информации: показатели, результаты или цифры, которые нужно наглядно представить, объяснить и проанализировать [4].

Дашборд представляет собой интерактивную информационную панель, которая демонстрирует аналитические данные. Пользовательский интерфейс

данного инструмента характеризуется одновременным выполнением нескольких функций и обновлением собранных сведений [30].

По своему назначению, а именно для того, чтобы позволить руководителю команды или менеджеру компании отслеживать ход работы, и с точки зрения методологии, информационная панель может быть определена как группа последовательных правил, методов, процедур, посредством которых осуществляется сбор информации для прогнозирования, мониторинга, оценки деятельности рабочей группы или организационных составляющих компании, в целом, а также для принятия решений, влияющих на продолжение рабочих активностей [26].

Существуют различные типы дашбордов, которые могут быть разработаны и использованы для мониторинга и визуализации данных. Рассмотрим некоторые из них:

- **Операционные дашборды:** Эти дашборды предоставляют операционную информацию о текущем состоянии системы, такую как производительность серверов, доступность сети, использование ресурсов и другие показатели. Они помогают операционным командам отслеживать и реагировать на проблемы в реальном времени.

- **Финансовые дашборды:** Для бизнеса важно отслеживать финансовые показатели, такие как выручка, прибыль, затраты и другие финансовые метрики. Финансовые дашборды предоставляют визуализацию этих показателей, позволяя руководству и финансовым отделам принимать информированные решения на основе данных [12].

- **Маркетинговые дашборды:** Дашборды маркетинга помогают компаниям отслеживать и анализировать эффективность своих маркетинговых кампаний и стратегий. Они могут включать данные о конверсии, трафике, рекламных расходах, показателях эффективности и других метриках маркетинговой деятельности.

- **Кадровые дашборды:** Эти дашборды предоставляют информацию о человеческих ресурсах компании, такую как уровень удовлетворенности

сотрудников, текучесть кадров, процесс найма, тренинги и другие HR-метрики. Они помогают HR-отделу и руководству принимать решения, связанные с управлением персоналом [7].

– Качественные дашборды: Эти дашборды предоставляют информацию о качестве продукта или услуги, например, количество ошибок, время простоя системы, удовлетворенность клиента и другие метрики. Они помогают командам по контролю качества и улучшению продукта принимать меры для устранения проблем и повышения качества.

– Проектные дашборды: Для управления проектами могут использоваться дашборды, которые отображают состояние проектов, сроки выполнения задач, ресурсы, бюджеты и другую связанную информацию. Они позволяют руководителям отслеживать прогресс, выявлять задержки и принимать меры для достижения поставленных целей.

Это лишь некоторые примеры типов дашбордов, которые могут быть использованы в IT-компаниях. Фактические типы дашбордов зависят от конкретных потребностей и целей компании.

Для отслеживания процессов тестирования мы работаем со следующими типами:

– «Операционный – отображает изменения данных в бизнесе. Примеры дашбордов для бизнеса – графики, с помощью которых можно посмотреть, как менялась посещаемость сайта и что на неё влияло» [10, 173].

– «Аналитический – помогает исследовать тенденции и делать выводы. Обычно их создают для конкретного бизнес-подразделения. Аналитики работают с ними, чтобы зафиксировать отклонения показателей и отследить причины. Пример – разработка дашборда об изменении числа пользователей конкретного продукта за неделю. С его помощью можно увидеть средний чек, долю клиентов и процент товарооборота по каждому из конкурентов» [10, 173].

– «Стратегический – нужен, чтобы составить представление о ситуации в целом или об отдельных показателях, выявляет проблемы и

помогает их исправлять. Например, создание дашборда о лояльности персонала поможет понять степень лояльности сотрудников и отследить её изменения среди разных групп» [10, 173].

Таким образом, дашборды представляют собой визуальные инструменты, позволяющие отслеживать и анализировать информацию о процессе тестирования в режиме реального времени. Они объединяют различные метрики и данные, предоставляя удобный способ отображения ключевых показателей производительности тестовой среды.

Вот некоторые ключевые характеристики дашборда:

- Визуализация данных: Дашборд предоставляет графическое представление данных в виде диаграмм, графиков, таблиц и других визуальных элементов. Это помогает быстро и наглядно понять текущее состояние и тренды важных показателей.

- Гибкость и настраиваемость: Дашборды обычно предоставляют возможность настройки и выбора нужных метрик и параметров, чтобы каждый пользователь мог получать информацию, соответствующую его потребностям и ответственности.

- Реальное время: Дашборды имеют возможность обновляться в режиме реального времени или с определенной периодичностью. Это позволяет оперативно отслеживать изменения и принимать управленческие решения на основе актуальных данных.

- Многомерность данных: Дашборды могут объединять данные из различных источников и предоставлять комплексный обзор производственных процессов, финансовых показателей, маркетинговых метрик и других ключевых аспектов деятельности компании.

- Интерактивность: Дашборды могут быть интерактивными, что позволяет пользователям взаимодействовать с данными, фильтровать их, проводить анализ и получать дополнительную информацию по требованию.

- Мобильная доступность: Современные дашборды часто предлагают мобильные приложения или адаптивный дизайн, позволяющий

просматривать и анализировать данные на различных устройствах, таких как смартфоны и планшеты.

– **Безопасность и конфиденциальность:** Дашборды обеспечивают механизмы защиты данных, контроля доступа и ролевой модели, чтобы гарантировать безопасность и предотвращать несанкционированный доступ к конфиденциальной информации.

– **Пользовательский опыт:** Качественные дашборды обладают интуитивно понятным пользовательским интерфейсом, который позволяет быстро ориентироваться и получать нужную информацию с минимальным количеством усилий.

Учитывая эти ключевые характеристики, дашборды становятся мощным инструментом для визуализации и мониторинга данных, а также для принятия управленческих решений.

В современном информационном мире, где данные играют ключевую роль в принятии стратегических решений, использование дашбордов становится неотъемлемой частью работы IT-компаний. Как подчеркнули Сарикайя и др., информационные панели стали ключевыми элементами поддержки принятия решений на основе данных [25].

В части интеграционного тестирования, которое является важным этапом разработки программного обеспечения, использование дашбордов может помочь в повышении эффективности и производительности [13].

Одним из главных преимуществ использования дашбордов является возможность визуализации данных. Дашборд предоставляет компактный и информативный обзор ключевых показателей производственных процессов и бизнес-метрик компании. Вместо того чтобы изучать большие объемы информации в виде таблиц и отчетов, сотрудники могут получать наглядное представление о текущей ситуации и трендах. Визуализация данных помогает лучше понять и анализировать информацию, что ведет к более обоснованным и быстрым решениям.

Панели мониторинга используются для различных целей, включая мониторинг важных показателей, предоставление обзора данных или для общения [25].

Визуализация помогает аналитикам понять свои данные, автоматизируя шаблоны для представления данных или предлагая интересные части данных [41]. На уровне отдельных диаграмм это помогает быстрее исследовать визуальные конструкции, которые лучше всего передают данные [40], а при просмотре нескольких диаграмм может помочь аналитикам быстро изучить различные аспекты и комбинации данных [42].

Помимо визуализации данных, дашборды уместно использовать для отображения ключевых показателей эффективности (от англ. «key performance indicators», KPI) компании, применимых к разным процессам. Можно рассчитать KPI для отдельных сотрудников, команд, отделов, филиалов и целых организаций.

Ключевые показатели эффективности могут быть представлены в качественном или числовом выражении. «Числовое выражение используется для параметров, которые можно измерить и рассчитать» [24]. «Качественное выражение используется для метрик, у которых нет принятых единиц измерения» [24]. В этом случае, оценивая процессы и эффективность работы оперируют следующими понятиями: «низкий», «высокий», «негативный», «положительный», «нейтральный», «средний».

Выделяют пять групп ключевых показателей эффективности, в зависимости от выбранных метрик:

- KPI результата – показывают, что и в каком количестве было достигнуто;
- KPI расходов – отображают, сколько ресурсов было израсходовано для достижения цели;
- KPI функционирования – демонстрируют насколько правильно была выполнена работа, соответствие процесса по достижению цели установленным стандартам, нормам и регламентам;

– KPI производительности – представляют собой оценку результативности, основанную на соотношении достигнутого результата к затраченному времени;

– KPI эффективности – вычисляют соотношение достигнутого результата к использованным ресурсам.

Как при разработке ключевых показателей эффективности, так и построении дашбордов в интеграционном тестировании можно использовать метрики для оценки процесса тестирования и оценки качества разрабатываемого продукта.

Рассмотрим примеры таких метрик:

– Покрытие кода: это метрика, показывающая, какая часть исходного кода была протестирована (например, путем измерения покрытия строк кода, функций или веток).

– Количество и процент выполненных тестов: указывает, сколько тестов было выполнено и какой процент от общего числа тестов это составляет.

– Время выполнения тестов: отслеживает время, затраченное на выполнение каждого теста или группы тестов, что может помочь в определении узких мест в процессе тестирования.

– Частота ошибок: показывает, сколько ошибок было обнаружено в процессе интеграционного тестирования, и какая часть всех тестовых прогонов они составляют.

– Время восстановления: измеряет время, затраченное на исправление ошибок после их обнаружения.

– Количество отложенных или отклоненных дефектов: эта метрика отслеживает число дефектов, которые были отложены или отклонены без исправления.

– Стабильность среды: измеряет, сколько раз тестовая среда становится недоступной из-за различных проблем (например, недоступность сервисов или непредусмотренные перезагрузки).

- Распределение дефектов по приоритету и критичности: показывает, как распределены дефекты по степени их важности и какие проблемы требуют наибольшего внимания.

- Время решения дефектов: измеряет время, затраченное на исправление дефектов после их обнаружения.

- Процент успешных прохождений тестов: указывает, какой процент тестов был успешно пройден без обнаружения ошибок или дефектов.

- Процент автоматизированных тестов: показывает, какой процент всех тестов был автоматизирован для ускорения и повышения эффективности процесса тестирования.

- Среднее время между деплоями: измеряет среднее время, прошедшее между развертываниями новых версий продукта, что может быть полезно для оценки скорости разработки и доставки.

- Частота сбоев: отслеживает, как часто происходят сбои или отказы во время интеграционного тестирования.

- Задержки в поставке: указывает, сколько разработанных функциональностей или исправлений было задержано в процессе интеграционного тестирования.

- Уровень удовлетворенности клиента: измеряет мнение клиента о качестве и процессе интеграционного тестирования, часто в форме опроса или обратной связи.

Это лишь некоторые из возможных метрик, которые могут быть использованы при построении дашбордов в тестировании. Выбор конкретных метрик зависит от целей, требований и контекста проекта.

Информационные панели позволяют отслеживать ключевые показатели эффективности компании в реальном времени. Это важно для контроля выполнения оценочных критериев, учета целей и управления производственными процессами.

Благодаря визуальной форме представления данных, проектный менеджер может мгновенно получать информацию о текущем состоянии

производства, увидеть ключевые показатели производительности, такие как объем выполненных задач, сроки их выполнения, загрузка команды и т.д. Это позволяет быстро реагировать на возникшие проблемы и предпринимать меры для их решения.

Кроме того, дашборды могут быть использованы для анализа производственной деятельности и выявления трендов. Собирая данные в единую систему, проектный менеджер может проводить анализ производительности, идентифицировать узкие места и находить пути для оптимизации процессов. Руководители рабочих команд могут подходить к информации, касающейся управления, в системном видении, что позволяет эффективно распределять человеческие, материальные, финансовые, информационные и временные ресурсы, более эффективно использовать рабочее время менеджеров, направляя их деятельность на решение ключевых проблем, стоящих перед компанией [26]. Это позволяет повысить эффективность работы компании и достичь более высоких результатов.

Также сотрудники могут самостоятельно отслеживать свои показатели и сравнивать их с поставленной задачей. Это помогает поддерживать фокус на результате и мотивирует команду к достижению лучших результатов.

Однако, чтобы панели мониторинга были действительно полезными инструментами управления, необходимо правильно собрать и представить данные. Проектный менеджер должен определить ключевые показатели производительности и их взаимосвязь, а также выбрать удобный формат представления данных. Важно помнить, что дашборды должны быть интуитивно понятными и легко читаемыми для всех участников команды.

Правильное использование информационных панелей при управлении производственными процессами может привести к улучшению результатов и повышению конкурентоспособности компании в современном рынке информационных технологий.

Выделим и перечислим некоторые преимущества использования дашбордов в IT-компаниях.

– Улучшение прозрачности данных имеет важное значение для повышения эффективности и принятия основанных на собранных данных решений. С помощью дашбордов ИТ-компании могут представлять информацию о производительности, состоянии проектов и других важных параметрах в реальном времени. Сотрудники могут получать доступ к этой информации и анализировать ее, что способствует более глубокому пониманию текущей ситуации.

– Дашборды обеспечивают доступность данных. Они помогают сотрудникам получать необходимую информацию легко и быстро, без необходимости обращаться к различным источникам. Это упрощает процесс принятия решений и дает возможность оперативно реагировать на изменения внешних условий. Благодаря централизованному хранилищу данных и интуитивно понятному интерфейсу, дашборды значительно упрощают доступ к информации и повышают эффективность работы сотрудников.

– Дашборды позволяют оперативно выявлять проблемы и анализировать результаты. Если какой-то KPI значительно отклоняется от установленных значений, сотрудники могут быстро принять меры для устранения проблемы. Это помогает предотвращать потери времени и ресурсов и обеспечивает более точное планирование и управление.

– Использование дашбордов позволяет мониторить важные параметры интеграционного тестирования, такие как время выполнения тестов, количество ошибок, покрытие кода и процент пройденных тестов. Благодаря этому, команда разработки может быстро обнаружить проблемные зоны и принять соответствующие меры.

– Дашборды позволяют в реальном времени отслеживать покрытие кода, количество ошибок и процент пройденных тестов. Это позволяет быстрее выявлять проблемные места и улучшать качество тестирования. Благодаря наглядной визуализации данных на дашбордах можно быстро принимать решения о необходимых изменениях в процессе тестирования.

– Дашборды позволяют выявить узкие места в процессе интеграционного тестирования и сфокусироваться на наиболее критичных аспектах. Путем анализа данных на дашбордах команда может определить, какие тесты требуют больше времени и ресурсов, и принять меры по оптимизации. Это помогает сократить время выполнения тестов и повысить общую производительность процесса.

– Дашборды могут быть связаны с различными системами и инструментами, что позволяет оптимизировать и автоматизировать многие процессы сбора данных. Это уменьшает ручной труд и повышает производительность. Например, можно настроить автоматическую выгрузку данных и обновление дашборда в режиме реального времени, что экономит время сотрудников на рутинных операциях.

– Дашборды улучшают взаимодействие между сотрудниками и способствуют увеличению прозрачности коммуникаций внутри рабочей группы, среди взаимодействующих команд и, в целом, внутри компании. Когда у всех сотрудников есть доступ к общим данным и KPI, это способствует лучшей коммуникации и сотрудничеству. Все участники проекта могут видеть актуальные данные и информацию, что помогает лучше понимать текущую ситуацию и принимать согласованные решения, а также снижает возможность недоразумений. Команды могут легко делиться информацией, обсуждать проблемы и искать решения на основе общих данных. Это повышает эффективность работы и улучшает командный дух.

Помимо этого, дашборды также могут служить важным инструментом для обратной связи и мотивации команды, поскольку позволяют членам команды видеть результаты своего труда и вносить свои предложения по улучшению производственных процессов.

Однако, необходимо учитывать и ограничения использования дашбордов. Во-первых, ошибочное или некачественное представление данных может привести к искажению информации и неправильному принятию решений. Важно убедиться в правильности и достоверности

источников данных, а также взвешенно анализировать информацию, прежде чем сделать выводы.

Кроме того, дашборды не всегда могут предоставить полный контекст данных, особенно в случае сложных и многомерных систем. Интерпретация и понимание данных может потребовать дополнительных знаний и навыков, чтобы адекватно оценить ситуацию.

Использование дашбордов в IT-компаниях имеет множество преимуществ, связанных с увеличением эффективности и производительности. Они помогают визуализировать данные, отслеживать KPI, повышать прозрачность и коммуникацию, быстро выявлять проблемы, а также автоматизировать и оптимизировать процессы. Информационные панели помогают в мониторинге и анализе результатов, улучшении взаимодействия команды, оптимизации времени и ресурсов, а также в повышении качества тестирования. Внедрение дашбордов может быть полезным шагом для команд, стремящихся к более эффективному интеграционному тестированию и достижению лучших результатов в разработке программного обеспечения. Однако, важно учитывать ограничения использования дашбордов и быть внимательными к качеству источников данных, а также адекватно анализировать информацию, прежде чем принимать решения.

1.3 Обзор существующих методов оптимизации дашбордов

Построение дашборда является трудоёмким процессом. Помимо выявленных KPI и метрик, необходимо выбрать наиболее удобный сервис, создать шаблоны, обучить сотрудников (руководителей команд, инженеров по проверке качества ПО и разработчиков) применять информационные панели для планирования своей деятельности, а также составить инструкции для правильной обработки данных, чтобы дашборды отображали только необходимые результаты тестирования.

Для исследования в данной работе используется уже созданный программный продукт Test Management System for Jira (tms.netcracker.com) – система управления тестированием на основе JIRA.

JIRA – коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами. Разработана компанией Atlassian, является одним из двух её основных продуктов (наряду с вики-системой Confluence). Имеет веб-интерфейс [17].

Дашборд приёмочных работ создаёт и организует руководитель команды. Предварительно создаётся проект, система меток, организуется необходимая фильтрация, загружаются все необходимые тестовые сценарии (мануальные и автотесты). В процессе приёмочных работ заводятся дефекты, вопросы, а также задачи на правки тестовых сценариев или документации.

Подробнее следует остановиться на системе меток и фильтрации. Инструмент Test Management System for Jira позволяет создавать директории для каждого конкретного проекта в отдельности, но в итоге, для постоянно работающих команд список проектов и задач становится неизмеримо велик настолько, что появляется потребность исключить из зоны видимости задачи других проектов. Для этого вводится система меток (labels), по которым затем можно настроить необходимые фильтры.

Важными метками могут быть:

- версия релиза, для которой проводится приёмочное тестирование (пример оформления такой метки: R.2022.winter, R.2022.summer);
- название и итерация принимаемого проекта (например: Proj15_iter4, Proj15_iter5);
- метка соответствия приёмочным работам (Acceptance_Proj15_iter4);
- метка соответствия этапу приёмочных работ (Validation, Smoke, Regression...);
- метка о принесённом тестовом сценарии (new, updated);

- метка принадлежности к автотестированию (ATs);
- метка о переходе к последующему этапу приёмочных работ (approved_for_smoke, approved_for_regression, not_for_smoke...), и другие.

В ходе работ список меток может расширяться.

Метки выставляются ситуативно на задачи или тестовые сценарии. Так, например, новый автотест на этапе валидации будет иметь обязательные метки версии релиза, итерация принимаемого проекта, метку соответствия приёмочным работам, метку о принесённом тестовом сценарии, метку принадлежности к автотестированию (R.2022.winter, Proj15_iter4, Acceptance_Proj15_iter4, Validation, new, ATs). А после успешного завершения валидации на такой автотест добавится ещё и метка approved_for_smoke. На мануальный тестовый сценарий никогда не получится поставить метку ATs, поскольку значение данной метки – именно принадлежность к автотестам (исключением является только ситуация, когда мануальный тестовый сценарий по итогам валидации решено автоматизировать прямо во время приёмочных работ).

Важно, чтобы все члены команд были осведомлены об используемых метках и максимально точно применяли их. Если метки применяются неточно, не в полной мере или не используются совсем, то появляется риск утраты важных задач из отображения в фильтрах и дашборде, что снижает эффективность использования графической информации, а также снижает качество информации о текущем состоянии работ.

Фильтры – это основа отображения графической информации в дашбордах. В фильтр заносится информация:

- о необходимых типах задач (например, Defect, QA Task, Question, Doc Task и другие),
- о статусе исполнения задачи (Open, In Progress, On Hold, Closed, Ready for Retest и другие),
- о приоритетах (Blocker, Critical, Major, Normal, Low),
- о системе меток,

- о родительском объекте (в случае, если нужно собрать производные задачи из-под главной, первично заведённой),
- и другие особенности, по которым следует производить сортировку всех задач в инструменте.

Для фильтров можно выставить уровень доступа по категориям пользователей системы Test Management System for Jira: только просмотр или просмотр с возможностью редактирования, – либо запретить просматривать всем, кроме создателя фильтра.

Дашборд строится из системы визуальных блоков (виджетов), каждый из которых транслирует информацию на основе конкретного фильтра. Визуальные блоки могут представлять собой все виды диаграмм, таблицы, списки, ленты времени, отчёты, демонстрирующие зависимость между задачами.

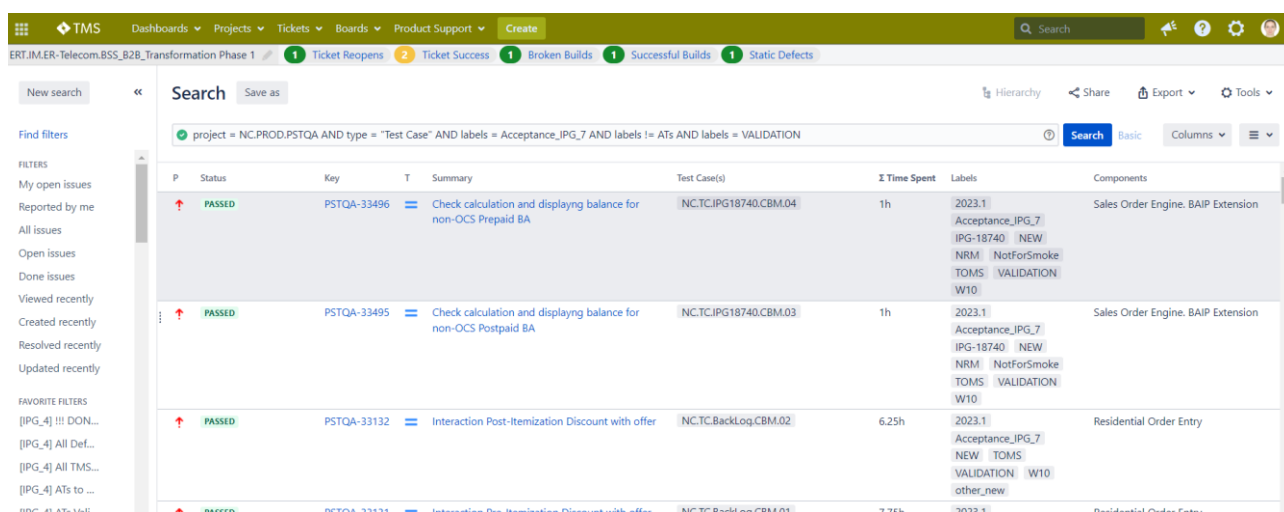


Рисунок 2 – Окно ввода фильтрации в Test Management System for Jira

Например, фильтр, отображающий все мануальные тестовые сценарии, для которых требуется провести валидацию (рисунок 2), выглядит так:

project = Proj15_iter4 AND type = "Test Case" AND labels = Acceptance_Proj15_iter4 AND labels != ATs AND labels = Validation

Two Dimensional Filter Statistics Plus					
Labels	Status				T:
	IN PROGRESS	CANCELLED	FAILED	PASSED	
new	1	16	1	46	64
updated	0	0	0	29	29
Total Unique Tickets:	1	16	1	75	93

Рисунок 3 – Скриншот виджета количества новых или изменённых тестовых сценариев и статус их прохождения

Данный фильтр может быть отображён в следующих видах:

- круговая диаграмма, описывающая количество тестовых сценариев по их приоритетам (рисунок 11),
- таблица, отражающая количество новых или изменённых тестовых сценариев и статус их прохождения (рисунок 3),
- таблица, распределяющая работы между членами команды и статус выполнения этих работ (рисунок 12).

Фильтр `type = Defect AND labels in (Acceptance_Proj15_iter4)`, отображающий общее количество дефектов, найденных во время приёмочных работ удобно разместить на дашборде в виде круговой диаграммы, описывающая количество дефектов по их приоритетам, либо в таблице, описывающей количество дефектов по отношению к этапам приёмочных работ (рисунок 4). А фильтр `type = Defect AND = Acceptance_Proj15_iter4 AND labels not in (Smoke, Regression) AND labels not in (not_a_bug, tested_passed)` содержит исключаяющие метки (`Smoke, Regression` и `tested_passed, not_a_bug`), по которым в отображение попадают только те валидационные дефекты, которые ещё в работе у разработчиков либо ожидают установки на тестируемое окружение и подтверждения о том, что дефект был поправлен. Т.е. если инженер по проверке качества ПО ставит метку о завершении проверки (`tested_passed`), то такой дефект исключается поиска для данного фильтра.

Two Dimensional Filter Statistics Plus								
Labels	Status							T:
	OPEN	CLOSED	ON HOLD	IMPLEMENTED	READY FOR TESTING	INFO RECEIVED	IN TECHNICAL REVIEW	
Regression	0	12	1	0	1	1	1	16
Validation	1	104	0	2	3	0	0	110
Smoke	1	32	0	1	1	0	0	35
Total Unique Tickets:	1	124	1	2	5	1	1	135

Рисунок 4 – Отображение общего количества дефектов

Filter Results:						
Key	T	Summary	P	Assignee	Due	Status
Short_ID_1	●	Name of defect 1	↑	Developer 1		OPEN
Short_ID_2	●	Name of defect 2	↑	Developer 2	12/Jan/23	IMPLEMENTED
Short_ID_3	●	Name of defect 3	↑	Developer 1	10/Jan/23	IMPLEMENTED

1-3 of 3

Рисунок 5 – Пример отображения списка дефектов, которые ещё в работе

Дефекты, которые ещё в работе, удобно отображать в виде списка (рисунок 5). Введение такого списка на дашборде позволяет оптимизировать временные затраты, и, вместо ежедневного просмотра всех заведённых дефектов, сконцентрироваться только на тех, которые актуальны и требуют внимания в данный момент.

В процессе работ информационные панели дополняются и расширяются новыми виджетами по запросу участников интеграционного тестирования. Поэтому случается так, что дашборды становятся слишком сложными и

перегруженными информацией, что затрудняет их эффективное использование. Важно систематически и своевременно пересматривать актуальные информационные панели на предмет соответствия целям проекта и рабочим задачам, корректировать фильтры и оптимизировать виджеты.

Во время пересмотра актуальных дашбордов следует задаваться следующими вопросами:

1. Какой остаётся цель и аудитория дашборда?

В данном случае необходимо подтвердить или переформулировать цель дашборда для интеграционного тестирования. Например, это может быть отслеживание производительности, выявление проблем в различных модулях или отображение текущего состояния проекта.

Также следует учитывать потребности аудитории, которая будет использовать дашборд. Информация, представленная на дашборде, должна быть понятной для пользователей, имеющих различный уровень компетенций и знаний о процессе тестирования программного обеспечения.

2. Остаётся ли прежним выбор ключевых метрик?

Можно разместить в верхних уровнях те виджеты, которые отражают наиболее важные метрики, соответствующие целям проекта. Например, это может быть количество успешно пройденных тестов, количество найденных ошибок, время реакции системы или уровень покрытия тестами.

Следует избегать перегрузки дашборда информацией. Для отображения должны быть выбраны только самые важные метрики, чтобы пользователи могли легко интерпретировать данные и принимать решения на их основе. Если метрики сменились, расширились/сократились, то это следует незамедлительно отразить и в дашбордах.

3. Что можно поменять в визуализации данных?

Оптимальный выбор типов виджетов поможет наиболее полно отразить собранные данные, а команде – быстро воспринять информацию и принять решения на основе анализа данных.

Как известно, в основу построения дашбордов ложатся фильтры. Один и тот же фильтр возможно представить в разных визуальных отображениях на дашборде. Виджет «Круговая диаграмма» будет наименее информативен, т.к. отображает данные о выборке фильтрации только по одному из выбранных параметров. Виджеты в виде таблиц более информативны, т.к. показывают данные о выборке фильтрации по двум параметрам. Также следует принимать во внимание виджет «Список», когда каждый из объектов фильтра требует детального вывода информации, но частое использование виджета со списком будет избыточно.

Следует уделять особое внимание выбору цветовой схемы и шрифтов. Они должны быть удобочитаемыми и не вызывать затруднений при просмотре дашборда.

Также следует проверить дашборд относительно существующих разделов или вкладок, чтобы организовать информацию по логическим группам и облегчить навигацию по дашборду.

4. Как отражено обновление и масштабирование?

Оптимизация дашборда включает в себя регулярное обновление данных. Следует предоставить возможность автоматического обновления данных в виджетах, чтобы пользователи всегда видели актуальную информацию.

Важно предоставить возможность настройки дашбордов в соответствии с актуальными потребностями конкретной команды. Например, команда разработчиков может заинтересована в отображении детальных данных об ошибках, в то время как менеджерам проекта может быть важно видеть общую статистику.

Необходимо учитывать, что дашборд может масштабироваться в связи с ростом проекта. Редактирование должно быть лёгким и позволять добавлять новые метрики и информацию по мере необходимости.

Оптимизация дашбордов для интеграционного тестирования играет важную роль в эффективности процесса разработки программного обеспечения. Выбор правильных метрик, подходящих типов визуализации и

настройка дашбордов в соответствии с потребностями команды позволяют проводить качественный и количественный мониторинг процесса тестирования. Оптимизированные дашборды предоставляют руководителю команды и отдельным сотрудникам проекта полезную информацию, помогают выявить проблемы и принять решения на основе анализа собранных данных.

Выводы по первой главе

Нами рассмотрены особенности приёмочного (интеграционного) тестирования и его значимость в части поддержания качества программного обеспечения. Описаны понятие дашборда и его роль в IT-компании, выявлены преимущества и возможности применения информационных панелей для принятия решений по управлению инженерами по тестированию. На примере обзора существующих методов оптимизации дашбордов выявлены актуальные подходы к повышению эффективности и функциональности информационных панелей.

Глава 2 Анализ опыта работы с дашбордами при приемочном тестировании в IT-компаниях

2.1 Анализ деятельности компании

Общество с ограниченной ответственностью «НетКрэкер» (рисунок 6) – «поставщик продуктовых решений для провайдеров услуг связи и кабельных операторов в мире. Компания специализируется на создании, внедрении и сопровождении систем поддержки бизнеса (Business Support System) и систем операционной поддержки (Operation Support System). Кроме этого, компания ведёт инновационные проекты в Cloud, B2B2X, Analytics/AI, LEO, поддерживает переход на 5G и технологию Open vRAN операторов связи по всему миру» [44].



Рисунок 6 – Лейбл компании Netcracker

Учредителем ООО «НетКрэкер» является Корпорация НетКрэкер Технологии Корпорэйшн (Netcracker Technology Corporation). Учредительный документ, на основании которого действует юридическое лицо ООО «НетКрэкер» – это Устав. Устав ООО «НетКрэкер» – это утвержденный в установленном порядке юридический документ, включающий свод положений и правил, касающихся правового статуса, организационной формы, структуры и устройства организации, видов деятельности, порядка отношений с юридическими и физическими лицами и государственными органами, а также определяющих права и обязанности как

участников организации, так и самого юридического лица. В уставе отражены основания учреждения организации, ее цели и задачи, описывается организационная структура как самой организации (наличие или отсутствие обособленных подразделений), так и органов ее управления, оговариваются форма собственности, порядок осуществления производственно-хозяйственной деятельности, а также правила реорганизации и ликвидации.

Полное фирменное наименование общества: Общество с ограниченной ответственностью «НетКрэкер» [3].

Сокращенное фирменное наименование общества: ООО «НЕТКРЭКЕР»

Организационно-правовая форма (ОПФ): Общество с ограниченной ответственностью.

Сведения о местонахождении общества: Юридический адрес - 115114, город Москва, Кожевническая улица, дом 7, строение 1, этаж 3, оф. 31.

Сведения о размере уставного капитала общества: 18 400 рублей.

Сведения о размере и номинальной стоимости доли каждого участника общества: Корпорация НетКрэкер Технолоджи Корпорэйшн (Netcracker Technology Corporation) является единственным участником. Её доля: 18 400 рублей.

ООО «НетКрэкер» является российским филиалом американской компании «Netcracker» и входит в японскую корпорацию NEC. «Специализируется на создании, внедрении и сопровождении систем поддержки бизнеса (Business Support System) и систем операционной поддержки (Operation Support System)» [44]. Компания обслуживает более 200 крупных организаций по всему миру.

Директором компании является Гроднева Галина Александровна. Она занимается непосредственным управлением.

Компания имеет филиалы в нескольких городах РФ, в том числе, в Самаре и Тольятти.

Сотрудники компании разделены на подразделения:

– бизнес-аналитики,

- продуктовые менеджеры,
- продуктовая разработка,
- специалисты по контролю качества и тестированию программного обеспечения,
- разработка группы поддержки,
- IT-специалисты (отдел отладки оборудования),
- специалисты учебного центра и другие.

Компания ООО «НетКрэкер» достигла высокого уровня информатизации, о чём свидетельствуют как количественные, так и качественные показатели.

Качественные характеристики:

- часть задач подлежит автоматизации,
- имеется интеграция на уровне комплексов функциональных задач и бизнес-процессов,
- имеются внутренние информационные ресурсы – корпоративная информационная система,
- развита IT-инфраструктура в масштабе предприятия,
- компания является лидером мировых рейтингов в области IT-решений и обладателем многих наград.

Количественные характеристики:

- процент автоматизированных задач от общего числа задач, подлежащих автоматизации – более 40%;
- все модули корпоративной информационной системы оснащены персональными компьютерами (1 компьютер на 1 сотрудника), мониторами (1-3 монитора на 1 сотрудника), точкой доступа в интернет, необходимым лицензионным программным обеспечением. Устаревшая техника своевременно заменяется на новые модели;
- практически все сотрудники являются сертифицированными IT-специалистами.

Финансовая информация также является количественной, характеристикой, но она не подлежит разглашению и не доступна рядовым сотрудникам.

Данная научно-исследовательская работа выполнялась в рамках прохождения практики в отделе по контролю качества и тестированию программного обеспечения в команде выдачи релизов.

Составим представление о процессах, предшествующих приёмочному (интеграционному) тестированию. ООО «НетКрэкер» разрабатывает программное обеспечение для «внедрения и сопровождения систем поддержки бизнеса и систем операционной поддержки» [44]. Функционал такого программного обеспечения зависит от конфигурации, запрошенной заказчиком. Конфигурация выстраивается из набора компонент программного обеспечения. Каждая компонента создаётся в отдельной команде, в которую входят проектные менеджеры, бизнес-аналитики, разработчики и инженеры по контролю качества, а затем выдаётся для приёмки в релиз. Релизы программного обеспечения формируются 4 раза в год. Команда релиза занимается приёмкой разработанных компонентов или обновлений к ним и последующим тестированием всех возможных версий релиза.

Основная работа в команде релиза заключена в подтверждении качества программного обеспечения в целом, качества принимаемых компонент и обновлений к ним, включённых в конфигурацию релиза, а также поддержании регрессионного списка тестовых сценариев (более 9 тысяч мануальных тестовых сценариев и более 3 тысяч автотестов) в актуальном состоянии.

Цель приемочного (интеграционного) тестирования заключается в проверке, насколько программное обеспечение функционирует надлежащим образом. При этом осуществляется проверка интеграции компонентов системы по критериям функциональности, производительности, безопасности и совместимости.

В приёмочные работы входят следующие активности:

- Изучение документации, предоставленной бизнес-аналитиками команды, выдающей обновления.
- Составление плана приёмочных работ (календарное планирование, планирование состава команды, планирование тестирования, составление инструкций принимающей команде).
- Оформление документации по приёмке, в том числе разработка отчётов, диаграмм и схем.
- Приёмочное тестирование:
 - а) валидационное тестирование,
 - б) смоук-тестирование,
 - в) регрессионное тестирование.
- Управление командой инженеров по тестированию во время приёмки.
- Описание итогов приёмки, проведение ретроспективы с принимающей командой и между командами (выдающей обновления и принимающей).

При успешном прохождении приемочного тестирования можно убедиться в работоспособности и соответствии разработанного продукта требованиям заказчика. Кроме того, это предоставляет возможность оптимизировать программное обеспечение или его отдельные компоненты с целью повышения их эффективности и удобства использования. Основная цель приемочного тестирования заключается в подтверждении общего качества программного обеспечения, а также качества всех принимаемых компонентов, включенных в конфигурацию релиза, и обновлений к ним. Дополнительно, это помогает поддерживать актуальность регрессионного списка тестовых сценариев.

Для оценки состояния приёмочного (интеграционного) тестирования используются сервисы системы JIRA, позволяющие создавать проекты (директории), список задач и дашборды [45]. Однако, традиционно, система дашбордов организована по вертикали вверх – это означает, что информация

передается от руководителя рабочей группы к менеджерам проектов и выше (рисунок 7) [46].

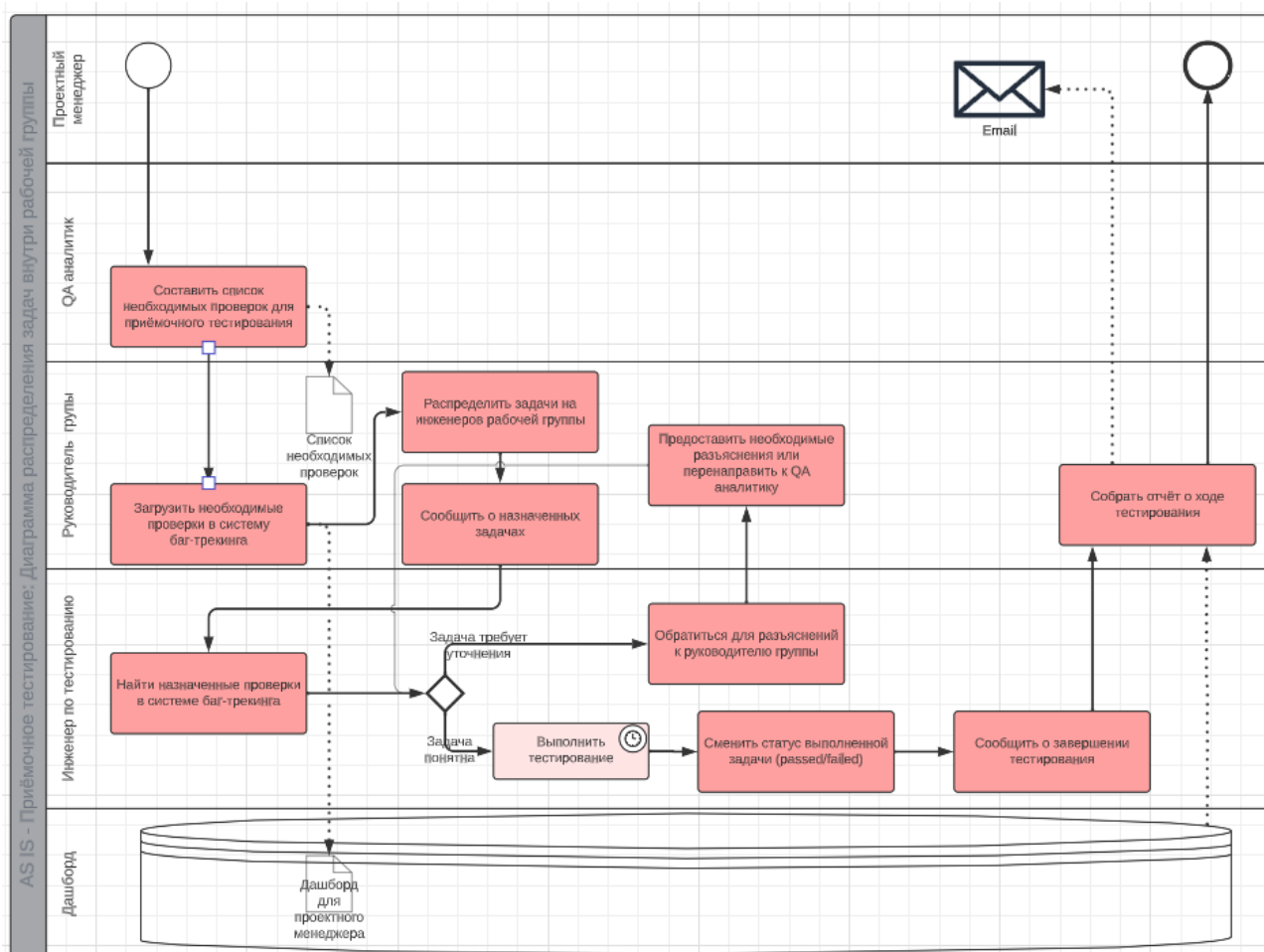


Рисунок 7 – AS IS – Приёмочное тестирование: Диаграмма распределения задач внутри рабочей группы

На диаграмме AS IS, в случае, когда дашборды организуются только для руководящего состава проекта, мы наблюдаем избыточность коммуникации между руководителем рабочей группы и инженерами по проверке качества программного обеспечения в части назначения рабочих задач и в части получения информации о том, что задача выполнена [43].

2.2 Использование дашбордов в процессе приемочного тестирования

В качестве средства реализации нами выбраны инструменты, предоставляемые системой JIRA, а именно Test Management System for Jira (TMS), т.к. именно эта система позволяет создать такой дашборд, который будет полезным для всех участников исполнения работ приёмочного тестирования и управленческого процесса, покажет рабочий процесс максимально наглядно и прозрачно [31].

JIRA настраивается индивидуально под потребности проекта, поэтому данный инструмент легко адаптировать к работе конкретной команды: можно выбрать стиль управления проектом, тип отображения задач, формат отчётов, вид дашбордов.

При работе с инструментами JIRA достаточно легко придерживаться итеративных методологий (Agile). Задачи в Jira помогают отслеживать всю работу на каждом этапе рабочего процесса вплоть до завершения. Jira помогает командам эффективно выполнять работу, действовать согласованно и общаться в контексте исполняемых задач.

Jira позволяет создавать задания: каждый пользователь может создать задачу для любого участника команды, добавить в описание задачи подробности, напоминания и даты выполнения. Большие задания можно разделить на несколько мелких подзадач. Те пользователи, которые участвуют в решении задачи, могут следить за её выполнением и оставаться в курсе дел, получая уведомления об изменении статуса работ по электронной почте. Все задания можно просматривать на дашборде, чтобы быстро определить состояние каждого из них [29].

Наличие полной информации о проекте позволяет создавать подробные отчеты об эффективности и ходе выполнения работы. Jira предоставляет необходимые инструменты для оценки, измерения и составления отчетов:

- Agile-отчеты: используются для анализа скорости работы команды, выявления узких мест и более точного прогнозирования будущей производительности;
- Отчеты DevOps: используются для анализа конвейера развертывания и частоты развертывания, чтобы повысить эффективность совместной работы и оптимизировать жизненный цикл продукта;
- Аналитические отчеты по задачам: используются для анализа того, на каких типах работ сосредоточена команда и как она справляется с работой;
- Прогнозные отчеты и отчеты об управлении: используются для оценки возможностей команды и более точного прогнозирования будущей производительности.

Также JIRA обладает способностью быть интегрированной с собственными и сторонними платформами и приложениями, например, решения для управления кодом и версиями; инструменты управления документацией; инструменты мониторинга и управления.

Таким образом, применение инструментов JIRA позволяет максимально прозрачно демонстрировать работу в команде, а также осуществлять преемственность наработанных материалов между сотрудниками в рамках одной итерации, а также между несколькими итерациями.

Руководитель рабочей группы по интеграционному тестированию формирует дашборд для приемочных работ. Сначала создается проект, система меток и фильтров. Затем загружаются все необходимые тестовые сценарии (мануальные и автоматизированные). В процессе проведения приемочных работ регистрируются задачи (tickets, «тикеты») различных типов: дефекты, вопросы, задачи на внесение изменений в тестовые сценарии или задачи на внесение изменений в документацию (рисунок 8).

Важно сделать дашборд полезным для всех участников приёмочного процесса, а также для проектного и продуктового менеджеров, и сделать рабочий процесс максимально наглядным и прозрачным через демонстрацию

актуального состояния работ. Такой принцип ведения дашборда обеспечивает не только хранение всей информации, доступной в любое время любому члену команды, а также позволяет обеспечивать преемственность дел между сотрудниками, если кто-то выходит из состава команды (больничные, отпуска, увольнения и т.д.). Информация, интересная менеджерам, – это общее количество имеющихся задач, демонстрация соответствия сроков их выполнения, отчёты, демонстрирующие особенные проблемные моменты. Информация для руководителей команд – количество актуальных задач, требующих решения, список оставшихся работ по этапам приёмочного процесса, распределение этих работ между всеми членами команды. Остальные участники приёмочного процесса также имеют доступ к дашборду, где могут отслеживать свои текущие задачи, приоритизацию их выполнения, статус выполнения задач другими коллегами, которые влияют на продолжение работ самого участника.

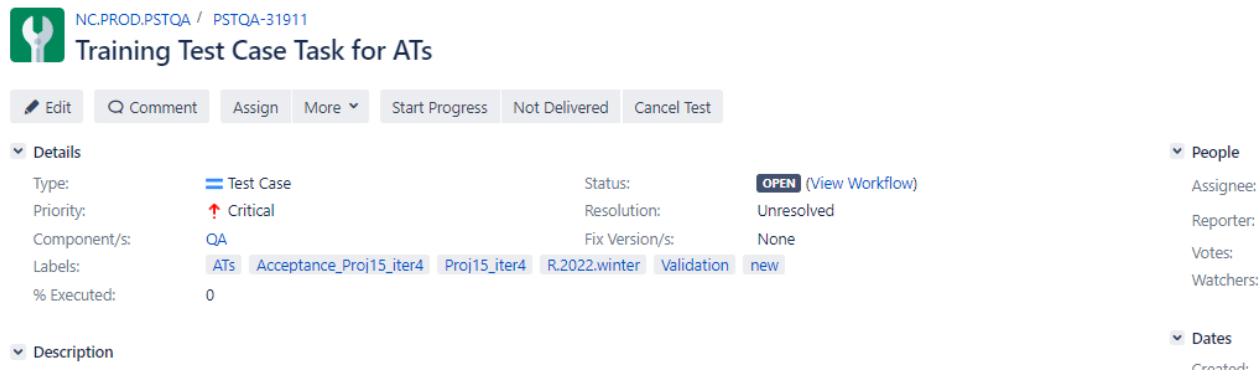


Рисунок 8 – Пример отображения задачи в TMS

В рамках первой итерации проекта Integration Product Gap (IPG_iter01-03) было проведено анкетирование инженеров по проверке качества программного обеспечения из рабочей группы по интеграционному тестированию (см. текст вопросов анкеты в Приложении А). Результаты первичного анкетирования отображены на рисунке 9.



Рисунок 9 – Результаты первичного анкетирования рабочей группы по тестированию.

Мы видим, что наиболее слабыми позициями в рабочих активностях инженеры считают недостаточно гибкое реагирование на изменения в целях или требованиях проекта, а также отсутствие возможности полноценного использования дашборда для отслеживания своих задач. Также первичный опрос показывает средний уровень удовлетворённости инженеров своими рабочими активностями в процессе тестирования.

В связи с вышеописанными проблемами руководитель практики от ООО «НетКрэкер» предложил следующее задание: «Разработать модель дашборда, содержащего актуальную и максимально концентрированную информацию о процессе приёмки, интересную всем участникам рабочего процесса и управления. Ввести модель дашборда в процесс приёмочного тестирования, апробировать на рабочей группе; при необходимости, оптимизировать.

Распространить применение данной модели на все внутренние проекты приёмочного тестирования».

Таким образом, бизнес-цель данного исследования заключается в создании такой модели дашборда, которая сократит коммуникации между руководителем рабочей группы и непосредственными исполнителями работ, при этом без потери скорости и качества приёмочного (интеграционного) тестирования. Пример модели см. на рисунке 10.

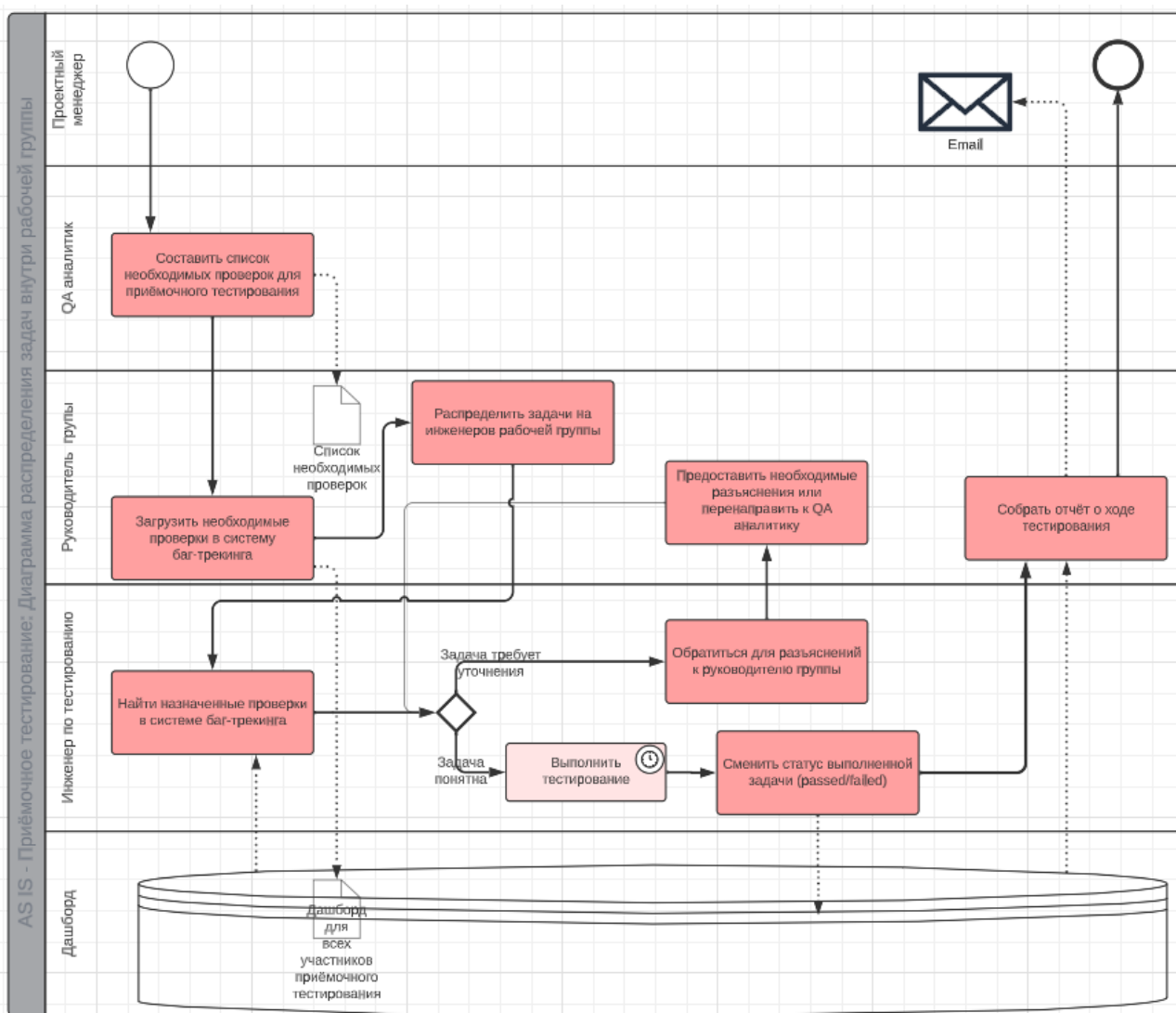


Рисунок 10 – TO BE – Приёмочное тестирование: Диаграмма распределения задач внутри рабочей группы.

На диаграмме AS IS (рисунок 9), в случае, когда дашборды организовываются только для руководящего состава проекта, мы наблюдаем избыточность коммуникации между руководителем рабочей группы и инженерами по тестированию в части назначения рабочих задач и в части получения информации о выполнении задачи.

Однако, если мы создаём дашборд для исполнителей, т.е. для инженеров по проверке качества программного обеспечения, часть задач подлежит автоматизации, то избыточные «Сообщить о назначении задач» и «Сообщить о завершении тестирования» исключаются из стандартных взаимодействий (рисунок 10). Таким образом, количество коммуникаций сокращается и сводится только к необходимым или исключительным случаям.

Преимущества и недостатки «дашборда для руководителей» (предполагается, что исполнители тестирования не будут его использовать) и «дашборда для всех участников рабочего процесса» (используется как управляющими, так и исполнителями) представлены в Приложении В.

Были составлены бизнес-план для внедрения новой модели дашборда (см. Приложение Б) и матрица ролей, определяющая участников этого процесса (таблица 2).

Таблица 2 – Матрица ролей

Задачи	Проектный менеджер	QA аналитик	Руководитель группы	QA инженеры
подобрать необходимые метрики	эксперт, который консультирует команду	ответственный за всю задачу	исполнитель задачи	должен быть в курсе выполнения задачи
описать фильтры на основе выбранных метрик	должен быть в курсе выполнения задачи	эксперт, который консультирует команду	ответственный за всю задачу	-
создать модель дашборда	должен быть в курсе выполнения задачи	эксперт, который консультирует команду	ответственный за всю задачу	должен быть в курсе выполнения задачи
обучить рабочую команду	должен быть в курсе	ответственный за всю задачу	исполнитель задачи	исполнитель задачи

Задачи	Проектный менеджер	QA аналитик	Руководитель группы	QA инженеры
пользоваться этим дашбордом	выполнения задачи			
распространить использование подобной модели на аналогичные процессы внутри компании ООО «НетКрэкер»	эксперт, который консультирует команду	ответственный за всю задачу; исполнитель задачи	помогает основному исполнителю	-

Разработанная модель бизнес-стратегии включила в себя следующие пункты:

- Анализ текущих процессов: Оценить текущие процессы и выявить основные проблемные области, связанные с временными затратами руководителей рабочих групп на административные работы и коммуникации с сотрудниками. Определить факторы, влияющие на уровень самоорганизации сотрудников.

- Разработать и внедрить дашборды: Создать модель дашборда, позволяющую руководителям и сотрудникам просматривать ключевые метрики и информацию в режиме реального времени. Дашборд должен быть интуитивно понятным, легким в использовании и предоставлять необходимую информацию для принятия решений на всех уровнях рабочего процесса.

- Обучение и поддержка: Предоставить необходимое обучение руководителям рабочих групп и сотрудникам по использованию дашбордов и освоению новых процессов, направленных на сокращение временных затрат и повышение самоорганизации. Обеспечить поддержку и обратную связь для эффективного использования дашбордов.

- Улучшить процессы: Провести мониторинг и оценку эффективности внедренных изменений. Данные с дашборда будут использоваться для анализа и определения улучшений в процессах. Внедрить новые методы и инструменты, основываясь на полученных данных и обратной связи от сотрудников и руководителей рабочих групп.

– Сократить время приёмочного тестирования: Разработать и реализовать стратегию по сокращению времени приёмочного тестирования. В нашем случае ожидается, что руководитель рабочей группы сможет уделять время непосредственно тестированию, что увеличит скорость тестирования без повышения количества сотрудников рабочей группы. Дополнительно это может включать оптимизацию процессов взаимодействия в рабочих группах, повышение эффективности коммуникации и сотрудничества между различными группами, вовлеченными в тестирование, автоматизацию тестирования.

– Непрерывное развитие и улучшение: Бизнес-стратегия должна быть динамичной и готовой к изменениям. Постоянно оценивать результаты, собирать обратную связь и вносить корректировки в стратегию, чтобы достичь желаемых целей и приспособиться к меняющимся бизнес-требованиям.

2.3 Анализ проблем, выявленных в опыте работы

Из существующих подходов к созданию дашбордов можно выделить проектный и продуктовый подходы [15]. Основное отличие между проектным и продуктовым подходами заключается в том, что первый ориентирован на текущее состояние проекта, а второй – на качество продукта в целом.

Для проведения приёмочного тестирования уместнее использовать проектный подход при создании дашборда. Дашборд, созданный в рамках проектного подхода, содержит информацию о текущем состоянии проекта, прогрессе выполнения задач и достижении целей проекта. Он будет включать в себя графики, диаграммы и другие элементы визуализации данных, а также возможности фильтрации и сортировки данных.

Построение дашборда уместнее планировать, исходя из метрик, ожидаемых к отображению. Для дашборда организации приёмочного тестирования были выбраны следующие метрики:

- Статус выполнения задач: текущий статус выполнения задач по тестированию, таких как создание тестовых случаев, выполнение тестов, анализ результатов и т.д.
- Количество успешных тестов: количество тестов, пройденных успешно.
- Количество неудачных тестов: количество сценариев, которые не прошли проверки.
- Количество пропущенных тестов: сценарии, которые не были запущены по каким-либо причинам.
- Количество компонент, покрытых тестами
- Время выполнения тестов: время, затраченное на выполнение автоматических и/или ручных тестов.
- Количество ошибок: количество ошибок, найденных во время тестирования приложения.
- Статус релиза: текущий статус релиза приложения, включая количество неисправленных ошибок и время до выпуска.

Первично разработанные дашборды были опробованы на 5 и 6 итерациях приёмочных работ проекта Integration Product Gap (IPG_iter5, IPG_iter6). Они показали недостаточную информативность по сравнению с ожидаемой. Также мы столкнулись с ограничениями инструмента Jira. В Test Management System for Jira (TMS) можно размещать не более 20 виджетов на одном дашборде. В следствие этого потребовалась оптимизация и пересмотр рабочего шаблона дашборда.

Нам потребовалось уменьшить количество ожидаемых виджетов и в то же время увеличить концентрацию информативной части о протекании рабочего процесса, т.к. при первичном построении шаблона мы столкнулись со следующими ограничениями при создании дашборда в Test Management System for Jira:

- ограниченный набор графических элементов для визуализации данных,

- ограниченные возможности настройки параметров отображения данных на дашборде,
- ограниченные возможности фильтрации данных по различным критериям,
- ограниченные возможности экспорта данных с дашборда для дальнейшего анализа.

Надежность и эффективность информационных панелей зависит от качества визуальных эффектов и данных, которые они представляют [28]. Исследования показывают, что менее четверти информационных панелей предоставляют информацию об их источниках, что является лишь одной из ожидаемых метаданных, если серьезно учитывать происхождение [23]. Происхождение – это запись, описывающая людей, организации, сущности и действия, которые сыграли роль в производстве, влиянии или доставке части данных или объекта [39].

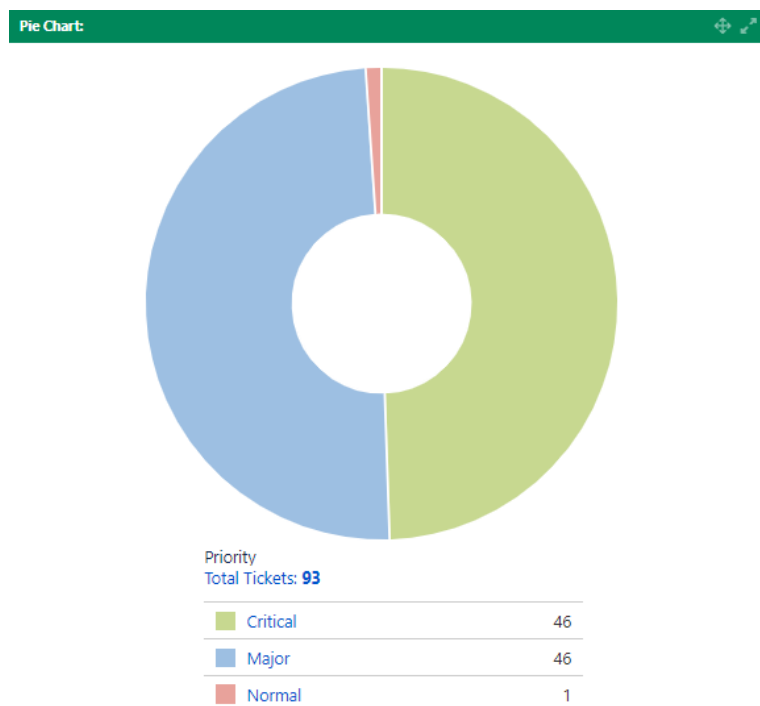


Рисунок 11 – Круговая диаграмма количества тестовых сценариев и их приоритетов

Как известно, в основу построения TMS дашбордов ложатся фильтры. Один и тот же фильтр возможно представить в разных визуальных отображениях на дашборде. Виджет «Круговая диаграмма» будет наименее информативен, т.к. отображает данные о выборке фильтрации только по одному из выбранных параметров (рисунок 11).

Виджеты в виде таблиц более информативны, т.к. показывают данные о выборке фильтрации по двум параметрам (рисунок 12).

Также следует принимать во внимание виджет «Список», когда каждый из объектов фильтра требует детального вывода информации (рисунок 13), но частое использование виджета со списком будет избыточно.

Two Dimensional Filter Statistics Plus					
Assignee	Status				T:
	IN PROGRESS	CANCELLED	FAILED	PASSED	
QA engineer 1	0	0	0	16	16
QA engineer 2	0	2	0	12	14
QA engineer 3	1	12	0	9	22
QA engineer 4	0	0	1	0	1
Team Lead	0	2	0	38	40
Total Unique Tickets:	1	16	1	75	93

Рисунок 12 – Распределение мануальных тестовых сценариев между инженерами на этапе валидационных работ

Filter Results:						
Key	T	Summary	P	Assignee	Due	Status
Short_ID_1	●	Name of defect 1	↑	Developer 1		OPEN
Short_ID_2	●	Name of defect 2	↑	Developer 2	12/Jan/23	IMPLEMENTED
Short_ID_3	●	Name of defect 3	↑	Developer 1	10/Jan/23	IMPLEMENTED

1-3 of 3

Рисунок 13 – Пример отображения списка дефектов, которые ещё в работе

В следствие этого потребовалась оптимизация и пересмотр рабочего шаблона дашборда:

- удалена большая часть виджетов с круговыми диаграммами, как наименее информативные. Вместо них добавлены табличные виджеты;
- скорректированы и расширены фильтры, собирающие информацию о дефектах и задачах на правку тестовых сценариев;
- учёт активных дефектов производится на единых виджетах, без деления по этапам приёмочных работ;
- учёт задач на правку тестовых сценариев производится на единых виджетах, без деления по этапам приёмочных работ;
- добавлен виджет учёта остаточных работ (Backlog) за все предыдущие итерации – такие работы планируются к выполнению в последующих итерациях;
- добавлен уточняющий виджет, позволяющий посчитать, сколько кейсов допущено в смоук тестирование.



Рисунок 14 – Результаты повторного анкетирования рабочей группы по тестированию

По итогам обучения и применения новой модели дашбордов, было проведено повторное анкетирование инженеров по проверке качества программного обеспечения из рабочей группы по интеграционному тестированию (см. текст вопросов анкеты в Приложении А). Результаты первичного анкетирования отображены на рисунке 14.

Мы видим, что слабые позиции в рабочих активностях инженеров устранены. Средний уровень удовлетворённости инженеров своими рабочими активностями в процессе тестирования повышается.

Выводы по второй главе

Мы провели анализ деятельности компании ООО «НетКрэкер» на примере организации приёмочного (интеграционного) тестирования, выявили потенциальные возможности для оптимизации данной активности. Нами выделены факторы, ограничивающие эффективность применения дашбордов, и предложены пути решения проблемы. Была разработана и апробирована первичная модель дашборда.

Глава 3 Разработка и оптимизация дашбордов для процесса приемочного тестирования

3.1 Проектирование дашбордов, учитывающих специфику процесса приемочного тестирования

Предыдущие этапы исследования продемонстрировали, что дашборд рассматривается как инструмент, предназначенный для передачи информации «вертикально вверх», т.е. от руководителя команды далее к менеджерам проектов и выше. Возможность использования дашборда непосредственными исполнителями работ не рассматривается, хотя является весьма актуальной с точки зрения практического применения.

Важно сделать дашборд полезным для всех участников исполнения работ и управленческого процесса, показать рабочий процесс максимально наглядным и прозрачным через демонстрацию актуального состояния работ. Такой принцип ведения дашборда обеспечивает не только хранение всей информации, доступной в любое время любому члену команды, а также позволяет обеспечивать преемственность дел между сотрудниками, если кто-то выходит из состава команды (больничные, отпуска, увольнения и т.д.). Информация, интересная менеджерам проекта, – это общее количество имеющихся задач, демонстрация соответствия сроков их выполнения, отчёты, демонстрирующие особенные проблемные моменты. Информация для руководителей команд – количество актуальных задач, требующих решения, список оставшихся работ по этапам приёмочного процесса, распределение этих работ между всеми членами команды. Остальным участникам приёмочного процесса также следует предоставить доступ к дашборду, где они смогут отслеживать свои текущие задачи, приоритизацию их выполнения, статус выполнения задач другими коллегами, которые влияют на продолжение работ самого участника.

[2023.1][Validation] IPG_7 - ATs early validation Dashboard

IPG_7[WH] All ATs - Two Dimensional Filter Statistics Plus

Labels	Status	In Progress	Closed	On Hold
NEW	4	4	4	4
To_Do	14	14	14	14
Total Unique Tickets:	18	18	18	18

Showing 2 of 2 statistics
Filter: [EQ_Ticket]@All

Filter Results: IPG_7[WH]@NEW

T	Key	Summary	P	Status	Assignee
1	PSTQA-32381	[PQ][BAP][Customer Billing Management@Release2023.1] Merge AT	+	IN ANALYSIS	Borys Merzhchi
2	PSTQA-32378	[PQ][BAP][Order Management@Release2023.1] Merge AT	+	IN ANALYSIS	Borys Merzhchi
3	PSTQA-32377	[PQ][BAP][LIMITED QUOTATION@Release2023.1] Merge AT	+	IN ANALYSIS	Borys Merzhchi
4	PSTQA-32376	[PQ][BAP][Product Offering Qualification@Release2023.1] Merge AT	+	IN ANALYSIS	Borys Merzhchi

1-4 of 4

Filter Results: IPG_7[WH]@RELEASED

T	Key	Summary	P	Status
1	PSTQA-32384	[PQ][AT_UPDATED][BAP][Monitoring@Release2023.1] Merge AT	+	IN ANALYSIS
2	PSTQA-32380	[PQ][AT_UPDATED][BAP][Order Management@Release2023.1] Merge AT	+	IN ANALYSIS
3	PSTQA-32382	[PQ][AT_UPDATED][BAP][Qualification@Release2023.1] Merge AT	+	IN ANALYSIS
4	PSTQA-32390	[PQ][AT_UPDATED][BAP][Product Offering Catalog@Release2023.1] Merge AT	+	IN ANALYSIS
5	PSTQA-32378	[PQ][AT_UPDATED][BAP][Customer Product Management@Release2023.1] Merge AT	+	IN ANALYSIS
6	PSTQA-32378	[PQ][AT_UPDATED][BAP][Product Offering Qualification@Release2023.1] Merge AT	+	IN ANALYSIS
7	PSTQA-32377	[PQ][AT_UPDATED][BAP][Customer Billing Management@Release2023.1] Merge AT	+	IN ANALYSIS
8	PSTQA-32376	[PQ][AT_UPDATED][BAP][Voucher Management@Release2023.1] Merge AT	+	IN ANALYSIS
9	PSTQA-32375	[PQ][AT_UPDATED][BAP][Order Management 2.0@Release2023.1] Merge AT	+	IN ANALYSIS
10	PSTQA-32374	[PQ][AT_UPDATED][Product Offering Catalog U@Release2023.1] Merge AT	+	IN ANALYSIS
11	PSTQA-32373	[PQ][AT_UPDATED][Product Offering Catalog@Release2023.1] Merge AT	+	IN ANALYSIS
12	PSTQA-32372	[PQ][AT_UPDATED][Product Lifecycle Management@Release2023.1] Merge AT	+	IN ANALYSIS
13	PSTQA-32371	[PQ][AT_UPDATED][Billing Activation and Information Provider@Release2023.1] Merge AT	+	IN ANALYSIS
14	PSTQA-32379	[PQ][AT_UPDATED][Residential Order Entry@Release2023.1] Merge AT	+	IN ANALYSIS

1-14 of 14

IPG_7[VALIDATION]@All - Two Dimensional Filter Statistics Plus

Labels	Cancelled	Failed	Passed	T
atns_updated	14	0	0	27
atns_exe	21	1	19	41
atns_updated	65	0	10	75
Total Unique Tickets:	100	1	29	130

Showing 3 of 3 statistics
Filter: [EQ_Ticket]@All

IPG_7[VALIDATION]@All - Two Dimensional Filter Statistics Plus

Assignee	NEW	To_Do	T
Georgii Iln	5	32	37
Ilya Petrov	0	3	3
Maryna Shumakova	6	29	35
Maryna Storchak	0	7	7
Sofia Lukina	0	0	0
Zarina Sharmova	30	18	48
Total Unique Tickets:	41	98	139

Showing 6 of 6 statistics
Filter: [EQ_Ticket]@All

IPG_7[VALIDATION]@All - Two Dimensional Filter Statistics Plus

Assignee	Cancelled	Failed	Passed	T
Georgii Iln	30	0	7	37
Ilya Petrov	2	0	1	3
Maryna Shumakova	37	0	0	37
Maryna Storchak	3	0	4	7
Sofia Lukina	4	0	4	8
Zarina Sharmova	34	1	14	49
Total Unique Tickets:	100	1	36	137

Showing 6 of 6 statistics
Filter: [EQ_Ticket]@All

IPG_7[VALIDATION]@All - Two Dimensional Filter Statistics Plus

Component	Cancelled	Failed	Passed	T
Billing Activation and Information Provider	2	0	1	3
Product Lifecycle Management	4	0	0	4
Product Offering Catalog	0	0	1	1
Product Offering Catalog UI	0	0	1	1
Residential Order Entry	0	0	0	0
Telecom Business API	00	1	29	30
Total Unique Tickets:	100	1	35	136

Showing 6 of 6 statistics
Filter: [EQ_Ticket]@All

IPG_7[ATs] Tickets - Two Dimensional Filter Statistics Plus

Labels	Open	In Progress	Closed	On Hold	T
VALIDATION	0	5	50	1	64
must_have_ATs	0	0	0	0	0
Total Unique Tickets:	0	5	50	1	79

Showing 2 of 2 statistics
Filter: [EQ_Ticket]@All

IPG_7[ATs] Tickets - Two Dimensional Filter Statistics Plus

Assignee	Open	In Progress	Closed	On Hold	T
Artem Rabbier	0	0	0	0	0
Borys Merzhchi	1	0	0	0	1
Georgii Iln	0	0	24	0	24
Ilya Petrov	0	0	1	0	1
Maryna Shumakova	0	5	0	0	5
Maryna Storchak	0	0	12	0	12
Olesya Durenchak	0	0	6	0	6
Zarina Sharmova	0	0	0	1	1
Total Unique Tickets:	0	5	58	1	79

Showing 8 of 8 statistics
Filter: [EQ_Ticket]@All

Filter Results: IPG_7[Active Item]@DEFECTS

Key	T	Summary	P	Assignee	Due	Status
IPG-34387	1	[QA_NC_RELEASE] Equipment Return Reason table doesn't show enough information	+	Hanna Tachya		READY FOR TESTING
IPG-33353	1	[QA_NC_RELEASE]@NC.TC.IP00-3463.003 CR Rollback from PROD server completed with errors	+	Sergei Maslaga	16/Jun/23	READY FOR TESTING
IPG-33006	1	[QA_NC_RELEASE] Money amount in RCE is not calculated correctly	+	Zarina Sharmova	14/Jun/23	READY FOR TESTING

1-3 of 3

Filter Results: IPG_7[Active Item]@DEFECTS

T	Key	Summary	Status	Assignee	Due	Status
1	PSUPFCM-81853	[QA_NC_RELEASE] Small font for column names	REQUEST ACCEPTED	Dyanna Shevina	21Jun23	REQUEST ACCEPTED
2	PSUPFCM-1775	[QA_NC_RELEASE] Metrics is unavailable on server	RESOLVED	Alansh Agapov	09Jun23	RESOLVED
3	PSUPFCM-4879	[QA_NC_RELEASE] Abbrev 'Phone Number' in customers Contact Methods doesn't use mask	TECHNICAL REVIEW	Sergey Chukhryng	23May23	TECHNICAL REVIEW
4	PSUPFCM-8095	[QA_NC_RELEASE]@IPG_3er7 Empty blank cells for Template Attachment in Flat Tables Import Configurations	RESOLVED	Mankanda Parasurama	26Jun23	RESOLVED
5	PSUPFCM-8056	[QA_NC_RELEASE] Price Alterations are not displayed after import Donecall	ASSESSMENT	Shagan Sharma	09Jun23	ASSESSMENT
6	PSUPFCM-8144	[QA_NC_RELEASE]@IPG_3er7 Empty blank cells for Template Attachment in Flat Tables Import Configurations	ASSESSMENT	Noukela Sheela	10Jun23	ASSESSMENT
7	PSUPFCM-8136	[QA_NC_RELEASE] Error importing flat file	IN QUEUE	Noukela Sheela	23Jun23	IN QUEUE
8	PSUPFCM-8139	[QA_NC_RELEASE] Cannot to remove flat filters	ADDITIONAL INFO REQ...	Anesha Daga	23Jun23	ADDITIONAL INFO REQ...
9	PSUPFCM-8122	[QA_NC_RELEASE] Manual changed price of NRC is not announced	ASSESSMENT	Rajakumar S	26Jun23	ASSESSMENT
10	PSUPFCM-8096	[QA_NC_RELEASE] Cannot to import flat file	ADDITIONAL INFO REQ...	Anesha Daga	16Jun23	ADDITIONAL INFO REQ...
11	PSUPFCM-8093	[QA_NC_RELEASE] Increased validation on same that Composite Filter Name 'is unique	RESOLVED	Anesha Daga	13Jun23	RESOLVED
12	PSUPFCM-8089	[QA_NC_RELEASE] Incorrect name of Billing Catalog Item on Excel file	TECHNICAL REVIEW	Noukela Sheela	16Jun23	TECHNICAL REVIEW

1-12 of 12

IPG_7[Active Item]@DEFECTS - Two Dimensional Filter Statistics Plus

Priority	Open	Closed	Ready for Testing	T
Bugfixer	0	7	0	7
Critical	0	24	4	28
Major	1	64	9	74
Normal	0	4	0	4
Total Unique Tickets:	1	99	13	113

Showing 4 of 4 statistics
Filter: [EQ_Ticket]@All

IPG_7[Active Item]@DEFECTS - Two Dimensional Filter Statistics Plus

Priority	Resolved	Closed	Technical Review	Assessment	Additional Info Required	In Queue	Request Accepted	T
Bugfixer	1	2	0	0	0	0	0	3
Critical	4	9	0	0	0	0	0	13
Major	1	69	1	2	2	0	0	77
Normal	1	3	1	1	0	0	1	7
Total Unique Tickets:	7	83	2	3	2	0	1	105

Showing 4 of 4 statistics
Filter: [EQ_Ticket]@All

Filter Results: IPG_7[WH]@All - Two Dimensional Filter Statistics Plus

T	Key	Summary	Status	Assignee
1	PSTQA-32377	[PQ][BAP][Release2023.1]@must_Have T8AP IPG_Regression	OPEN	Borys Merzhchi
2	PSTQA-32376	[PQ][POC][Release2023.1]@must_Have PLM.POC.Smoke Multi-Import_Catalog IPG_Regression	OPEN	Borys Merzhchi
3	PSTQA-32375	[PQ][POC][Release2023.1]@must_Have PLM.POC.AM.Equipment_Offering_creation_reservation_end_checking IPG_Regression	OPEN	Borys Merzhchi
4	PSTQA-32374	[PQ][M2M][Release2023.1]@must_Have PLM.M2M.Voucher_Management_must_have IPG_Regression	OPEN	Borys Merzhchi
5	PSTQA-32373	[PQ][ROE AC][Release2023.1]@must_Have Asset_Management_with_BOIplus IPG_Regression	OPEN	Borys Merzhchi
6	PSTQA-32372	[PQ][ROE][Release2023.1]@must_Have Flat_Structures IPG_Regression	OPEN	Borys Merzhchi
7	PSTQA-32371	[PQ][BAP][Release2023.1]@must_Have BAP IPG_Regression	OPEN	Borys Merzhchi
8	PSTQA-32362	[PQ][POC][Release2023.1]@must_Have PLM.POC.Smoke Import_Flat_Entities IPG_Regression	OPEN	Ilya Petrov

1-8 of 8

IPG_7[Must Have ATs] Tickets (cancelled) - Two Dimensional Filter Statistics Plus

Labels	Open	Cancelled	T
BAP	1	0	1
Flat-Import	4	0	4
Flat-Structures	0	1	1
IPG_Backlog	5	0	5
Multi-Import	1	0	1
Total Unique Tickets:	6	1	7

Showing 4 of 4 statistics
Filter: [EQ_Ticket]@All

IPG_7[Must Have ATs] Tickets (cancelled) - Two Dimensional Filter Statistics Plus

Assignee	Open	Cancelled	T
Borys Merzhchi	0	0	0
Ilya Petrov	5	1	6
Total Unique Tickets:	5	1	6

Showing 2 of 2 statistics
Filter: [EQ_Ticket]@All

IPG_7[ATs] Tickets: Prod_TEAM - Two Dimensional Filter Statistics Plus

Assignee	Closed	On Hold	T
Artem Rabbier	2	0	2
Georgii Iln	7	0	7
Maryna Storchak	0	0	0
Olesya Durenchak	0	0	0
Zarina Sharmova	1	1	2
Total Unique Tickets:	10	1	11

Showing 5 of 5 statistics
Filter: [EQ_Ticket]@All

IPG_7[ATs] Tickets: Acceptance_TEAM - Two Dimensional Filter Statistics Plus

Assignee	Open	In Progress	Closed	T
Artem Rabbier	0	0	4	4
Borys Merzhchi	1	0	0	1
Georgii Iln	0	0	17	17
Ilya Petrov	0	0	1	1
Maryna Shumakova	0	5	0	5
Maryna Storchak	0	0	0	0
Zarina Sharmova	0	0	0	0
Total Unique Tickets:	0	5	26	31

Showing 7 of 7 statistics
Filter: [EQ_Ticket]@All

Рисунок 15 – Оптимизированный дашборд на валидацию автотестов

Нами разработана модель дашборда, позволяющая вывести рабочую команду проверки качества ПО на уровень самоорганизации в части получения и приоритизации рабочих задач, но в то же время сохраняющая элементы для отчётов вышестоящим руководителям [9].

Мы сохранили ключевые метрики для дашборда организации приёмочного тестирования:

- статус выполнения задач,
- количество успешно пройденных тестов,
- количество неудачных тестов,
- количество пропущенных сценариев,
- количество компонент ПО, покрытых проверками,
- время выполнения всех тестовых сценариев (автоматических и мануальных),
- количество ошибок, найденных во время тестирования.
- статус релиза.

Оптимизированный шаблон дашборда содержит следующие элементы:

- таблица количества Work Items на приёмку новых автотестов или приёмку изменений для уже имеющихся автотестов;
- список Work Items на приёмку новых автотестов;
- список Work Items на приёмку изменений для уже имеющихся автотестов;
- таблица количества новых или изменённых автотестов, разделённых по ключевым категориям, с демонстрацией результатов валидации;
- таблица распределения автотестов на инженеров приёмочных работ;
- таблица покрытия компонент предложенными автотестами;
- таблица количества задач, заведённых на правку автотестов, на всех этапах приёмочных работ;

- таблица распределения задач, заведённых на правку автотестов, на инженеров приёмочных работ;
- список Work Items для работ с автотестами из must_have_ATs группы;
- таблица покрытия компонент автотестами из must_have_ATs группы;
- таблица распределения задач на изменения автотестов из must_have_ATs группы;
- круговая диаграмма, отображающая количество автотестов, допущенных на мёрж в реестр регрессионных автотестов Релиза;
- таблица распределения задач на мёрж автотестов, на инженеров приёмочных работ;
- круговая диаграмма, отображающая количество автотестов, допущенных в смоук;
- таблица распределения автотестов, допущенных в смоук, на инженеров приёмочных работ;
- круговая диаграмма количества регрессионных автотестов и результатов (статусов) запусков таких автотестов;
- таблица распределения регрессионных автотестов на инженеров приёмочных работ;
- таблица общего количества дефектов, заведённых в рамках приёмочных работ на команду проектных разработчиков;
- таблица общего количества дефектов, заведённых в рамках приёмочных работ на команду поддержки (Support);
- список актуальных дефектов, заведённых в рамках приёмочных работ на команду проектных разработчиков;
- список актуальных дефектов, заведённых в рамках приёмочных работ на команду поддержки (Support);

- таблица количества новых или изменённых мануальных тестовых сценариев, разделённых по ключевым категориям, с демонстрацией результатов валидации;
- таблица распределения мануальных тестовых сценариев на инженеров приёмочных работ;
- таблица покрытия компонент предложенными мануальными тестовыми сценариями;
- таблица количества задач, заведённых на правку мануальных тестовых сценариев, на всех этапах приёмочных работ;
- таблица распределения задач, заведённых на правку мануальных тестовых сценариев, на инженеров приёмочных работ;
- круговая диаграмма, отображающая количество мануальных тестовых сценариев, допущенных в смоук;
- таблица распределения мануальных тестовых сценариев, допущенных в смоук, на инженеров приёмочных работ;
- круговая диаграмма количества регрессионных мануальных тестовых сценариев и результатов (статусов) их тестирования;
- таблица распределения регрессионных мануальных тестовых сценариев на инженеров приёмочных работ;
- таблица количества задач, заведённых на правку документации, на всех этапах приёмочных работ;
- таблица количества перенесённых задач (Backlog) по всем предыдущим итерациям приёмки;
- список задач перенесённых из текущей итерации приёмочных работ;
- виджет, формирующий репорт зависимостей автотестов от задач и дефектов, заведённых в рамках валидации;
- виджет, формирующий репорт зависимостей мануальных тестовых сценариев от задач и дефектов, заведённых в рамках валидации;

- круговая диаграмма количества автотестов, ожидаемых для включения в смоук-тестирование;
- таблица реального количества автотестов, попавших в смоук-тестирование;
- круговая диаграмма количества мануальных тестовых сценариев, ожидаемых для включения в смоук-тестирование;
- таблица реального количества мануальных тестовых сценариев, попавших в смоук-тестирование;
- виджет с временными границами приёма автотестов;
- виджет с временными границами приёма мануальных тестовых сценариев.

Поскольку количество виджетов превышало 20, нам потребовалось разделить эту информацию на 5 дашбордов по смысловым задачам каждого из этапов приёмочных работ (рисунки 15-19):

- дашборд на валидацию автотестов;
- дашборд на валидацию мануальных тестовых сценариев;
- дашборд на смоук и регрессию автотестов;
- дашборд на смоук и регрессию мануальных тестовых сценариев;
- вспомогательный дашборд с отчётами и временными границами.

Дашборды валидационного тестирования (рисунок 15, рисунок 16) наиболее наполнены по сравнению с другими, потому что данный этап приёмочного тестирования наиболее продолжителен, по сравнению с другими. Этот этап включает множество аспектов, таких как проверка функциональности, безопасности, производительности и охватывает широкий спектр данных, связанных с различными видами тестов и результатами проверок.

[2023.1][Validation] IPG 7 - manual TCs early validation Dashboard

[IPG_7][Validation][TCs] Testing - Two Dimensional Filter Statistics Plus

Labels	Cancelled	Failed	Passed	Passed with minor defects	Ready for retest	T
other_jira	0	0	13	2	2	4
other_updated	0	0	3	1	0	4
Back_new	36	1	60	0	0	103
Back_updated	0	0	5	0	0	5
Total Unique Tickets:	36	1	129	3	2	154

Showing 4 of 4 statistics
Filter: [IPG_7][Validation][TCs] Testing

[IPG_7][Validation][TCs] Testing - Two Dimensional Filter Statistics Plus

Components	Cancelled	Failed	Passed	Passed with minor defects	Ready for retest	T
Billing Activation and Information Provider	0	0	3	0	1	4
Product Lifecycle Management	4	0	3	0	0	7
Product Lifecycle Management - RBM integration	0	0	1	0	0	1
Product Offering Catalog	1	0	11	0	0	12
Product Offering Catalog UI	2	0	5	0	0	10
RBM Product Offering Catalog extension	4	7	10	0	0	14
Residential Order Entry	7	0	12	2	1	22
Sales Force Automation	0	0	7	0	0	7
Sales Order Engine	2	0	0	0	0	2
Sales Order Engine - BAIP Extension	0	0	2	0	0	2
Sales Order RBM extension	0	0	1	0	0	1
Telcom Business JSP	36	1	65	0	0	102
Total Unique Tickets:	66	1	129	2	2	196

Showing 12 of 12 statistics
Filter: [IPG_7][Validation][TCs] Testing

[IPG_7][Validation][TCs] Testing - Two Dimensional Filter Statistics Plus

Assignee	Cancelled	Failed	Passed	Passed with minor defects	Ready for retest	T
Georgii Ilin	0	0	4	0	0	4
Ira Pervov	5	0	10	0	1	16
Maryna Shumilova	6	0	10	0	0	16
Maryna Storchaka	5	0	39	0	0	44
Natalia Kostyleva	2	0	0	0	0	2
Nina Kolliva	0	0	0	0	1	1
Olesya Donsaevskaya	0	0	6	3	0	9
Sergii Tsyganenko	20	0	32	6	0	58
Sofia Luntina	7	0	7	0	0	14
Zarina Shamsova	11	1	14	0	0	26
Total Unique Tickets:	66	1	129	9	2	196

Showing 10 of 10 statistics
Filter: [IPG_7][Validation][TCs] Testing

[IPG_7][TCs] Tickets - Two Dimensional Filter Statistics Plus

Labels	Open	Reopened	Resolved	Closed	Additional info required	T
REGRESSION	1	0	1	1	0	3
VALIDATION	0	1	2	52	1	56
Total Unique Tickets:	1	1	3	53	1	59

Showing 2 of 2 statistics
Filter: [IPG_7][TCs] Tickets

[IPG_7][TCs] Tickets - Two Dimensional Filter Statistics Plus

Assignee	Open	Reopened	Resolved	Closed	Additional info required	T
Dmitrii Filigov	0	0	0	1	0	1
Georgii Ilin	0	0	0	5	0	5
Ira Pervov	1	0	1	6	0	8
Maria Onishina	0	0	0	1	0	1
Maryna Shumilova	0	0	2	0	0	2
Maryna Storchaka	0	0	2	0	0	2
Natalia Kostyleva	0	0	0	19	0	19
Nina Kolliva	0	0	0	1	0	1
Olesya Donsaevskaya	0	0	0	9	0	9
Sergii Tsyganenko	0	1	0	1	0	2
Sofia Luntina	0	0	2	0	0	2
Vadym Turv	0	0	0	1	0	1
Vladimir Dergachev	0	0	0	0	1	1
Zarina Shamsova	0	0	1	7	0	8
Total Unique Tickets:	1	1	4	55	1	62

Showing 14 of 14 statistics
Filter: [IPG_7][TCs] Tickets

[IPG_7][TCs] Tickets Prod. TEAM - Two Dimensional Filter Statistics Plus

Assignee	Open	Resolved	Closed	Additional info required	T
Dmitrii Filigov	0	0	1	0	1
Georgii Ilin	0	0	4	0	4
Ira Pervov	1	1	6	0	8
Maria Onishina	0	0	1	0	1
Maryna Shumilova	0	0	2	0	2
Maryna Storchaka	0	0	17	0	17
Olesya Donsaevskaya	0	0	8	0	8
Vadym Turv	0	0	1	0	1
Vladimir Dergachev	0	0	0	1	1
Zarina Shamsova	0	1	4	0	5
Total Unique Tickets:	1	2	44	1	48

Showing 10 of 10 statistics
Filter: [IPG_7][TCs] Tickets Prod. TEAM

[IPG_7][TCs] Tickets Acceptance. TEAM - Two Dimensional Filter Statistics Plus

Assignee	Reopened	Resolved	Closed	T
Georgii Ilin	0	0	2	2
Maryna Storchaka	0	0	2	2
Natalia Kostyleva	0	0	1	1
Nina Kolliva	0	0	1	1
Olesya Donsaevskaya	0	1	0	1
Sergii Tsyganenko	0	0	1	1
Sofia Luntina	0	2	0	2
Zarina Shamsova	0	0	3	3
Total Unique Tickets:	0	2	11	14

Showing 8 of 8 statistics
Filter: [IPG_7][TCs] Tickets Acceptance. TEAM

[IPG_7][TCs] Analysis - Two Dimensional Filter Statistics Plus

Assignee	Critical	Major	Normal	Low	T
Georgii Ilin	0	0	0	1	1
Ira Pervov	3	1	0	3	7
Maryna Shumilova	5	0	0	0	5
Maryna Storchaka	7	4	2	1	14
Natalia Kostyleva	0	0	0	1	1
Olesya Donsaevskaya	2	3	0	6	11
Sofia Luntina	0	1	0	0	1
Vladislav Hrytkov	0	0	0	1	1
Zarina Shamsova	9	0	0	2	11
Total Unique Tickets:	26	8	2	15	51

Showing 9 of 9 statistics
Filter: [IPG_7][TCs] Analysis

Filter Results [IPG_7] Active Items DEFECTS

Key	T	Summary	P	Assignee	Due	Status
IPG-33553	●	[QA_NC_RELEASE][NC_TC_IPG-34634.03] CR Rollback from PROD server completed with errors	↑	Sergii Masagin	16/Jun/23	READY FOR TESTING
IPG-34787	●	[QA_NC_RELEASE] Equipment Return Reason table doesn't show enough information	↑	Hanna Tachyla		READY FOR TESTING
IPG-33006	●	[QA_NC_RELEASE] Money amount in ROE is not calculated correctly	↑	Zarina Shamsova	14/Jun/23	READY FOR TESTING

1-3 of 3

[IPG_7] Active Items DEFECTS - Two Dimensional Filter Statistics Plus

Labels	Status	T
REGRESSION	READY FOR TESTING	2
Total Unique Tickets:		2

Showing 1 of 1 statistics
Filter: [IPG_7][Active Items DEFECTS]

Filter Results [IPG_7] Active Items DEFECTS

T	Key	P	Summary	Status	Due
●	IPSPUBTJ41653	↓	[QA_NC_RELEASE] Small font for column names	REQUEST ACCEPTED	21/Jun/23
●	IPSPUBTAP1775	↑	[QA_NC_RELEASE] Metrics is unavailable on server	RESOLVED	09/Jun/23
●	IPSPUBM4679	↑	[QA_NC_RELEASE] Attribute "Phone Number" (in customers Contact Methods) doesn't use mask	TECHNICAL REVIEW	23/May/23
●	IPSPPOC4095	↓	[QA_NC_RELEASE][SPO_Inv7] Empty blank cells for Template Attachment in Flat Tables Import Configurations	RESOLVED	28/Jun/23
●	IPSPPOC4056	↑	[QA_NC_RELEASE] Price Alterations are not displayed after Import Discount	IN QUEUE	09/Jun/23
●	IPSPPCBM4144	↓	CLONE - [QA_NC_RELEASE][SPO_Inv7] Empty blank cells for Template Attachment in Flat Tables Import Configurations	ASSESSMENT	10/Jun/23
●	IPSPPCBM4136	↑	[QA_NC_RELEASE] Error importing flat filter	IN QUEUE	23/Jun/23
●	IPSPPCBM4130	↑	[QA_NC_RELEASE] Cannot to remove flat filters	ADDITIONAL INFO REQUIRED	22/Jun/23
●	IPSPPCBM4122	↑	[QA_NC_RELEASE] Manual changed price of NRC is not overridden	ASSESSMENT	20/Jun/23
●	IPSPPCBM4095	↑	[QA_NC_RELEASE] Cannot to import flat filter	ADDITIONAL INFO REQUIRED	16/Jun/23
●	IPSPPCBM4093	↑	[QA_NC_RELEASE] Incorrect validation on same flat "Composite Filter Name" is unique	RESOLVED	13/Jun/23
●	IPSPPCBM4080	↓	[QA_NC_RELEASE] Incorrect name of "Billing Catalog Name" on Excel file	TECHNICAL REVIEW	10/Jun/23

1-12 of 12

[IPG_7] All Items DEFECTS - Two Dimensional Filter Statistics Plus

Priority	Open	Closed	Ready for testing	T
● Blocker	0	7	0	7
↑ Critical	0	24	4	28
↑ Major	1	64	9	74
↓ Normal	0	4	0	4
Total Unique Tickets:	1	95	13	113

Showing 4 of 4 statistics
Filter: [IPG_7][All Items DEFECTS]

[IPG_7] All Items DEFECTS - Two Dimensional Filter Statistics Plus

Status	Blocker	Critical	Major	Normal	T
RESOLVED	1	4	1	1	7
CLOSED	2	9	10	3	24
TECHNICAL REVIEW	0	0	1	1	2
ASSESSMENT	0	0	1	1	2
ADDITIONAL INFO REQUIRED	0	0	2	0	2
IN QUEUE	0	0	2	0	2
REQUEST ACCEPTED	0	0	0	1	1
Total Unique Tickets:	3	13	17	7	40

Showing 7 of 7 statistics
Filter: [IPG_7][All Items DEFECTS]

[IPG_7] Items DOC DEFECTS - Two Dimensional Filter Statistics Plus

Assignee	Resolved	Closed	T
Georgii Ilin	0	4	4
Ira Pervov	0	16	16
Ivan Kostov	1	0	1
Maryna Storchaka	0	3	3
Olga Anshangskaya	0	1	1
Sergii Tsyganenko	0	1	1
Sofia Luntina	1	0	1
Zarina Shamsova	0	1	1
Total Unique Tickets:	2	28	30

Showing 8 of 8 statistics
Filter: [IPG_7] Items DOC DEFECTS

[IPG_7] Active Backlog for all iterations - Two Dimensional Filter Statistics Plus

Labels	Open	In Progress	Reopened	Resolved	Closed	On Hold	Ready for testing	Cancelled	Blocked	Failed	Passed	DE
IPG_31_33_Backlog	0	0	0	3	3	1	0	0	0	0	0	11
IPG_4_Backlog	1	0	0	0	1	0	2	0	0	0	0	4
IPG_5_Backlog	0	0	0	2	1	0	2	1	0	2	0	8
IPG_6_Backlog	3	1	1	1	3	2	0	0	3	0	0	14
IPG_7_Backlog	4	1	0	0	0	3	3	0	0	2	4	17
Total Unique Tickets:	8	2	1	6	8	6	5	3	4	4	4	53

Showing 5 of 5 statistics
Filter: [IPG_7] Active Backlog for all iterations

[IPG_7][SPO] Analysis - Two Dimensional Filter Statistics Plus

Assignee	Open	In Progress	Resolved	Closed	On Hold	T
Georgii Ilin	0	0	0	7	0	7
Ira Pervov	0	3	0	2	2	7
Maryna Shumilova	2	3	0	0	0	5
Maryna Storchaka	0	0	14	0	0	14
Natalia Kostyleva	0	1	0	0	0	1
Olesya Donsaevskaya	4	0	1	4	2	11
Sofia Luntina	0	0	1	5	1	7
Vladislav Hrytkov	0	0	1	0	0	1
Zarina Shamsova	0	1	0	10	0	11
Total Unique Tickets:	6	8	2	43	5	64

Showing 9 of 9 statistics
Filter: [IPG_7][SPO] Analysis

Рисунок 16 – Оптимизированный дашборд на валидацию мануальных тестовых сценариев

Дашборд, отражающий статус прохождения смоук-тестирования и регрессии автотестов (рисунок 17), наполнен незначительно, т.к. после валидации, проведённой должным образом, на этапе смоук-тестирования не ожидаются дефекты и какие-либо ещё проблемы. Автоматические тестовые сценарии передаются в команду автотестирования для последующего мёржа принимаемых сценариев в реестр регрессионного тестирования релиза.

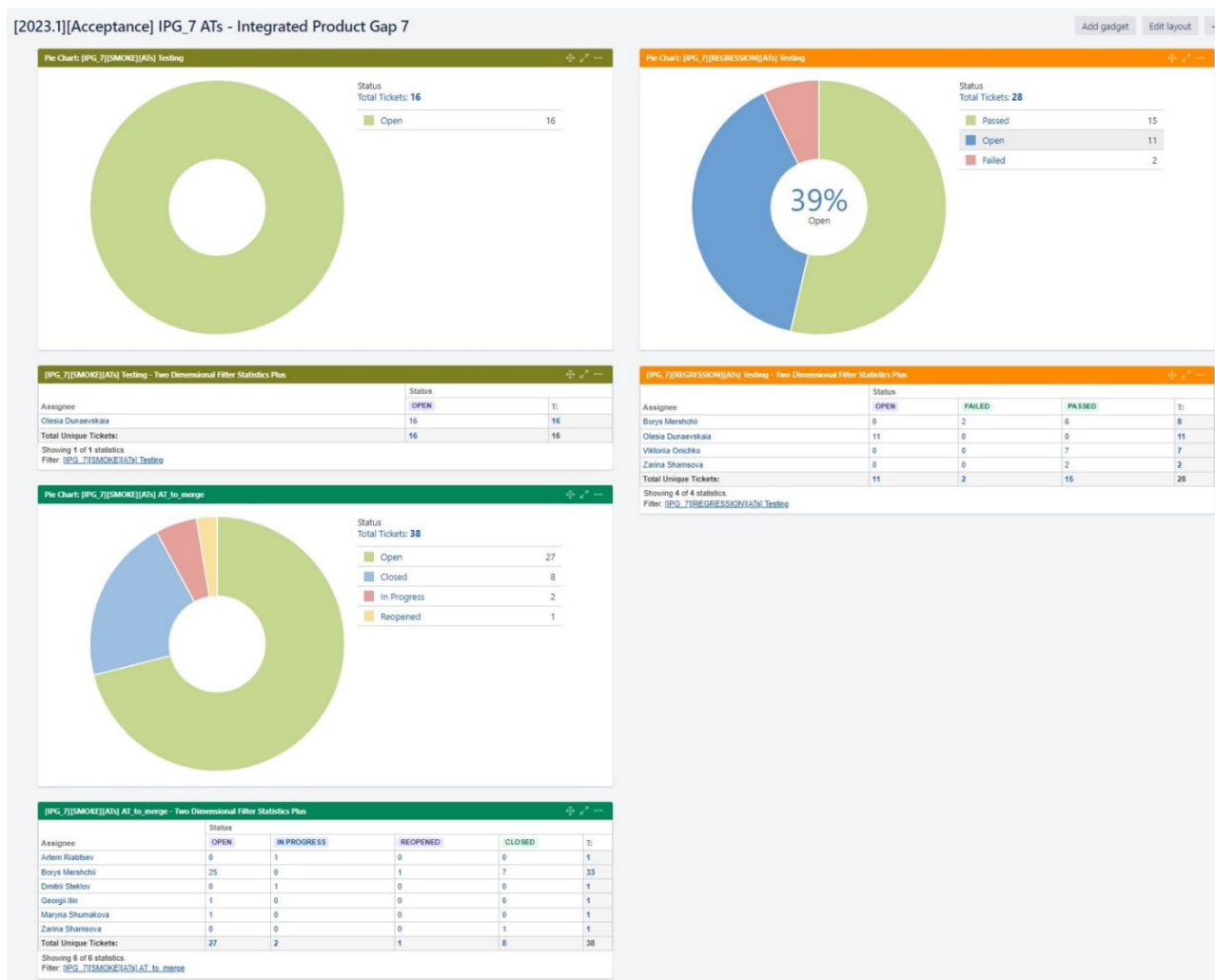


Рисунок 17 – Дашборд на смоук и регрессию автотестов в 7 итерации

Дашборд, отражающий статус прохождения смоук-тестирования и регрессии мануальных тестовых сценариев (рисунок 18), заполнен более подробно, т.к. мануальное тестирование предусматривает больше нюансов в проверках, по сравнению с автотестами, и здесь могут появляться

незначительные дефекты низких приоритетов, которые тем не менее также нуждаются в отслеживании и внесении исправлений.

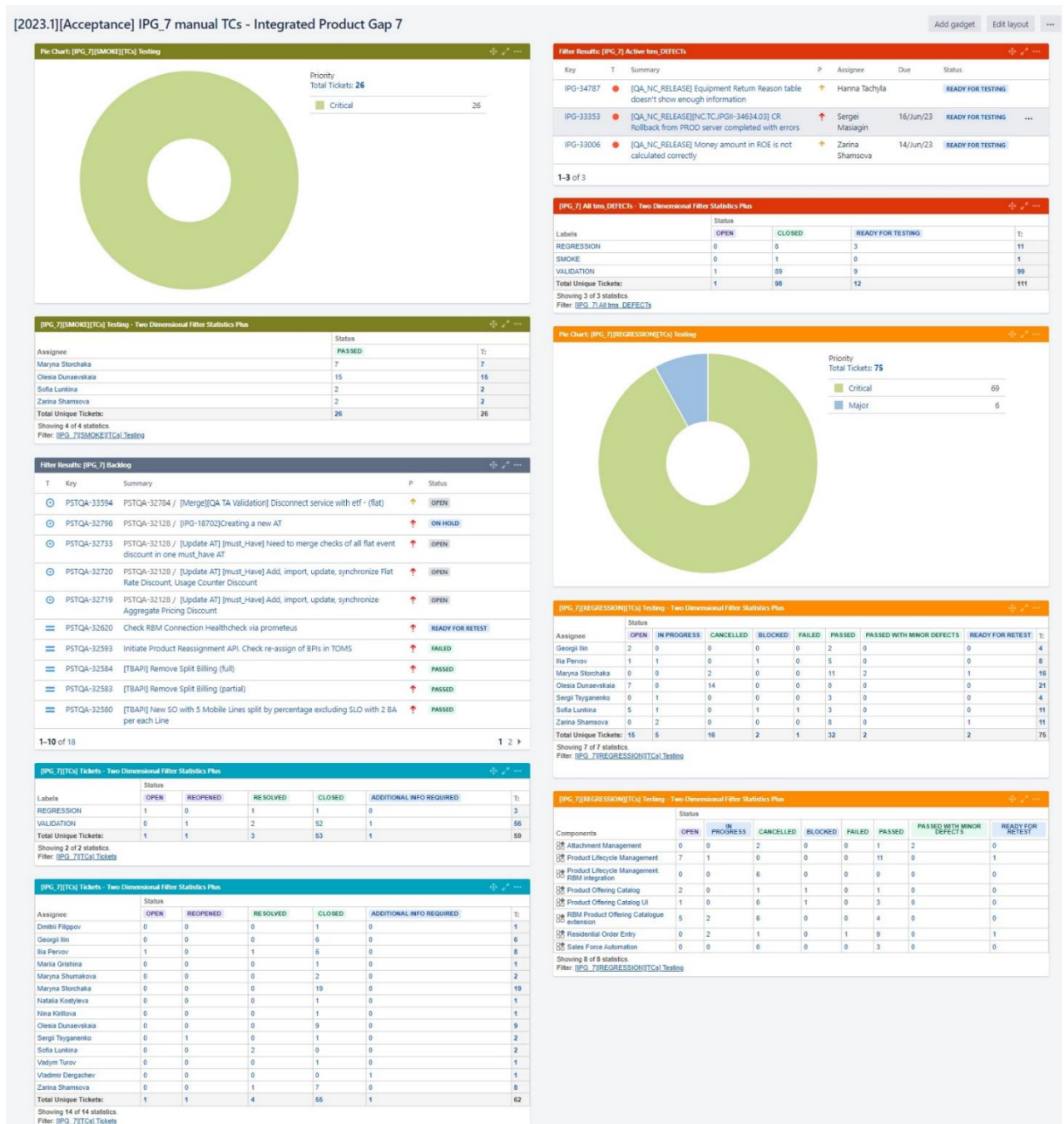


Рисунок 18 – Оптимизированный дашборд на смоук и регрессию мануальных тестовых сценариев

Новым для итоговой модели дашбордов, которые мы рекомендуем к применению в приёмочном (интеграционном) тестировании является

вспомогательный дашборд (рисунок 19). Слева он отражает процесс приёмочных работ, связанных с автотестами: фильтры, необходимые для репортов, «ленту времени» проведения автотестирования, количество автоматических тестовых сценариев и таблицу с отложенными автотестами (backlog), написание которых планируется в следующих итерациях. И по аналогии, справа – аналогичные виджеты, относящиеся к проведению приёмочных работ с мануальными тестовыми сценариями.

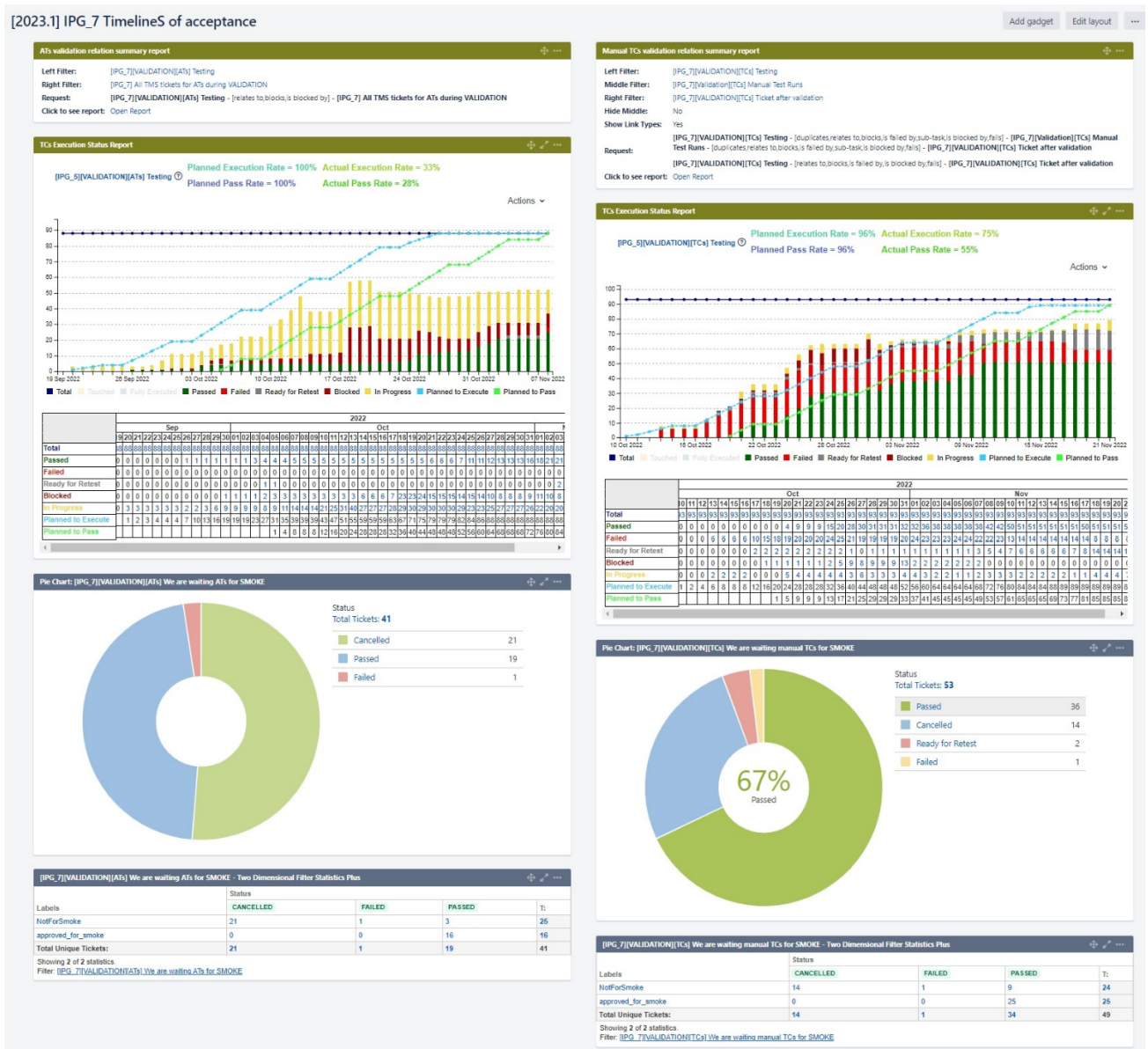


Рисунок 19 – Оптимизированный дашборд для репортов и вспомогательных виджетов

Для обучения рабочей группы инженеров интеграционного тестирования применению данной модели дашбордов написана статья на внутренних ресурсах ООО «НетКрэкер», а также проведены установочные сессии до начала приёмочных работ.

Итоговое решение было протестировано на 7 итерации приёмочных работ проекта Integration Product Gap (IPG_iter7).

3.2 Оценка эффективности разработанных дашбордов

Для проведения эксперимента на основе ранее описанной гипотезы используется следующая модель оценки:

- Измерить базовые показатели. Перед началом эксперимента собрать базовые данные о рабочей группе проверки качества программного обеспечения: о самостоятельности в получении рабочих задач, их приоритизации, целеполагании и ориентировании в рабочем процессе.

- Внедрение дашбордов. Описать метрики, создать фильтры на их основе. Создать дашборд. Внедрить дашборды. Обучить участников группы пользоваться дашбордами для получения рабочих задач, их приоритизации, целеполагания и ориентирования в рабочем процессе.

- Мониторинг и измерение. Следить за процессом работы группы, использующей новую модель дашборда и собирать данные, согласно заявленным критериям наблюдений. Измерить показатели производительности и эффективности. Провести опрос с целью получения обратной связи от сотрудников.

- Анализ результатов. Провести анализ данных, чтобы определить, есть ли значимая разница между результатами в рабочей группе при использовании новой модели дашборда. Сравнить результаты с бизнес-целями проекта.

- Выводы и интерпретация. Оценить результаты эксперимента и сделать выводы о том, как применение новой модели дашбордов влияет на

рабочую группу по проверке качества программного обеспечения. Определить, улучшилось ли целеполагание, ориентирование в рабочем процессе и производительность у рабочей группы.

Нами выделены следующие параметры для оценки эффективности работы сотрудников:

- Производительность: Измерение времени, затрачиваемого на выполнение задач при использовании дашбордов и без них. Сравнение производительности сотрудников до и после внедрения дашбордов.

- Качество работы: Оценка качества выполненных задач и ошибок, допущенных до и после внедрения дашбордов. Сравнение уровня ошибок и качества работы сотрудников в обеих группах.

- Самоорганизация: Измерение уровня самоорганизации сотрудников в группе, использующей дашборды, и в группе, не использующей их. Опрос сотрудников, чтобы оценить их способность самостоятельно получать задачи и приоритизировать их.

- Целеполагание: Оценка способности сотрудников устанавливать цели и выстраивать планы действий с использованием дашбордов. Сравнение уровня целеполагания сотрудников до и после внедрения дашбордов.

- Ориентирование в рабочем процессе: Измерение уровня ориентирования сотрудников в процессе работы с использованием дашбордов. Оценка их способности понимать текущие задачи, приоритеты и контекст работы.

Рассмотрим динамику эксперимента на примере итераций проекта «Integration Product Gap»:

- IPG_01-03
- IPG_4
- IPG_5
- IPG_6
- IPG_7

В части производительности и качества работ мы наблюдаем сокращение сроков приёмочного тестирования (таблица 3) при сокращении количества членов рабочей группы (рисунок 20).

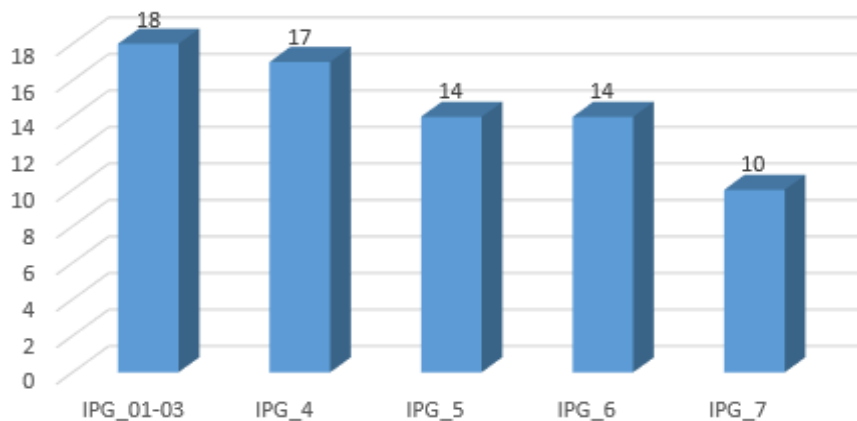


Рисунок 20 – Диаграмма. Количество человек, задействованных в приёмочном тестировании итераций проекта Integration Product Gap

Мы наблюдаем, что члены рабочей группы, непосредственные исполнители тестирования, были сокращены с 18 до 10 человек.

Рассмотрим таблицу сроков приёмочного тестирования (таблица 3) и переведём полученные значения в столбчатую диаграмму (рисунок 21). Мы наблюдаем динамику сокращения рабочего времени. С каждой новой итерацией на приёмочное тестирование уходило всё меньше времени.

Таблица 3 – Сроки приёмочного (интеграционного) тестирования для итераций проекта

Проект	Дата начала приёмки	Дата завершения приёмки	Всего дней
IPG_01-03	09.02.2022	07.07.2022	148
IPG_4	15.06.2022	20.10.2022	127
IPG_5	20.09.2022	30.01.2023	132
IPG_6	15.12.2022	07.03.2023	82
IPG_7	06.03.2023	19.05.2023	74

Мы считаем, что применение рекомендуемой модели из 5 дашбордов позволило сократить затраты на коммуникации между руководителем группы и инженерами по проверке качества программного обеспечения, что в пересчёте на общую длительность приёмочного процесса высвобождает значительные временные ресурсы.

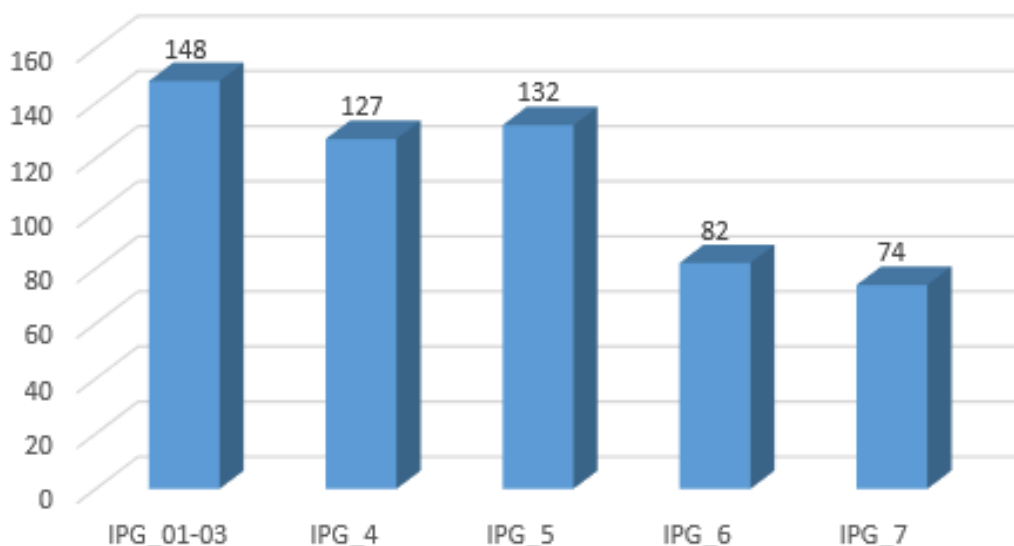


Рисунок 21 – Диаграмма. Количество дней, затраченных на приёмочное тестирование итераций проекта Integration Product Gap

Также следует отметить динамику повышения количества пройденных тестовых сценариев на одного инженера (рисунок 22).

Первично сниженное количество обработанных тестовых сценариев можно обосновать освоением и обучением в рамках нового подхода к использованию дашбордов. Однако линия тренда явно указывает на положительную динамику процесса.

В части самоорганизации мы наблюдаем сокращение времени общих совещаний (рисунок 23), а также снижение количества обращений к руководителю группы для распределения рабочих заданий (рисунок 24).

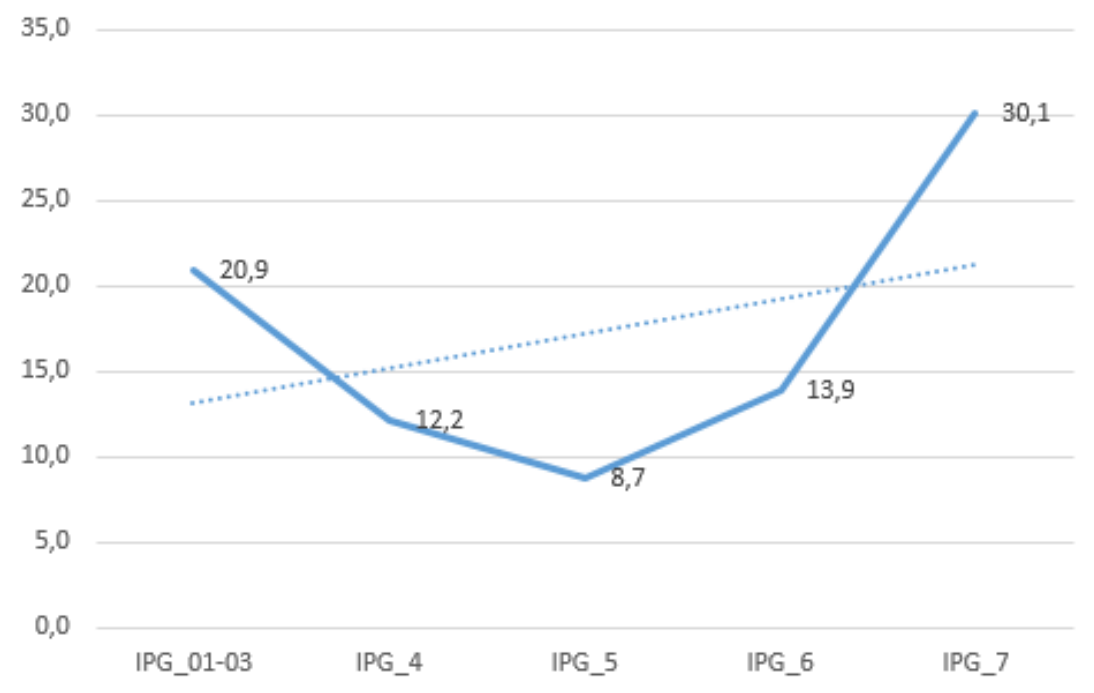


Рисунок 22 – Диаграмма: Среднее количество обработанных тестовых сценариев на одного инженера.

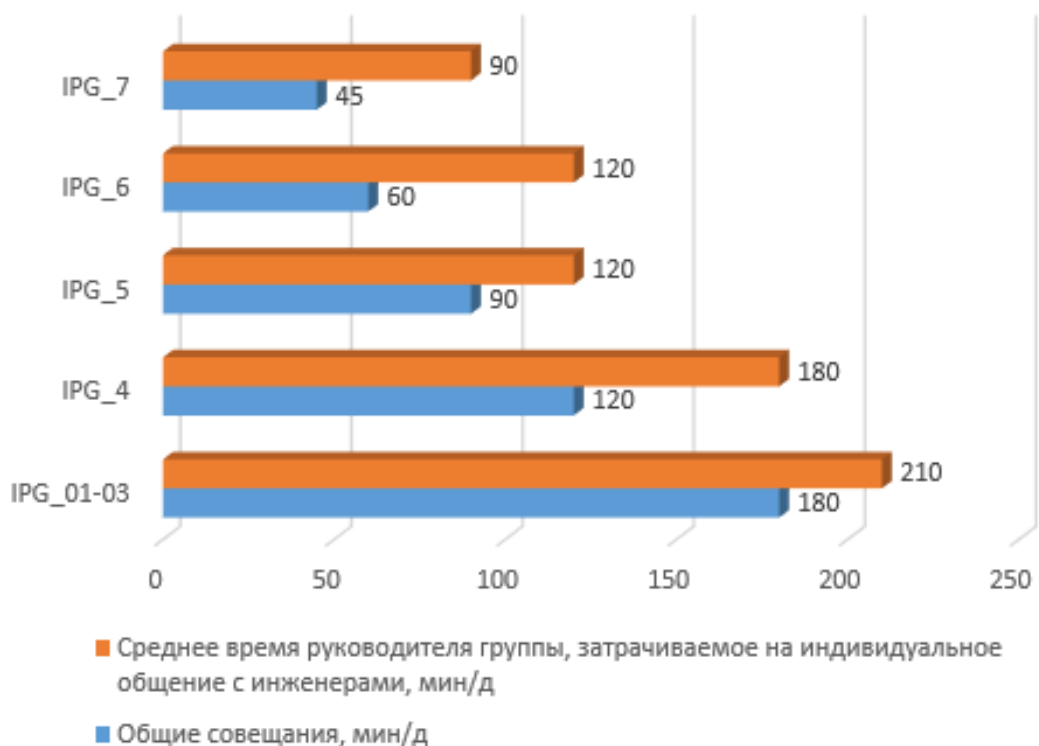


Рисунок 23 – Диаграмма времени, затраченного на коммуникации с членами рабочей группы

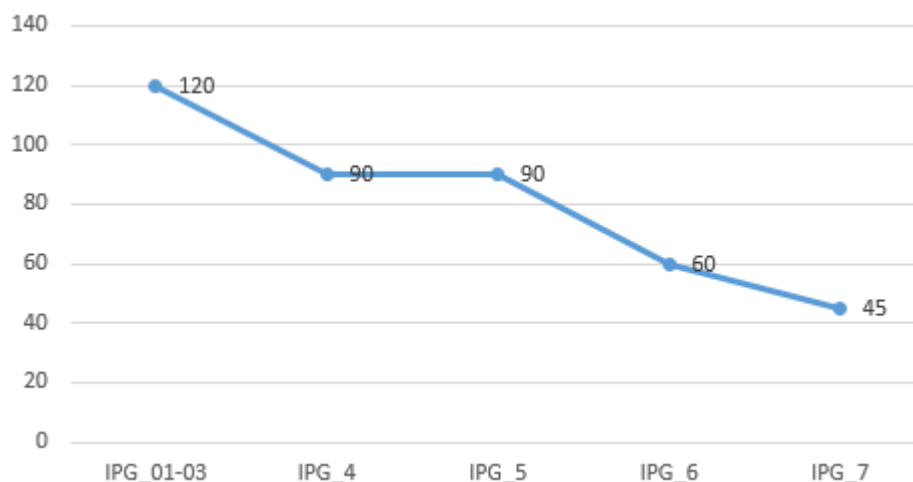


Рисунок 24 – График времени (в минутах), затраченного на построение отчётов

Для проверки уровня целеполагания и ориентирования в рабочем процессе было проведено анкетирование (Приложение А) сотрудников на итерациях IPG_1-3, IPG_5 и IPG_7 (рисунок 25).

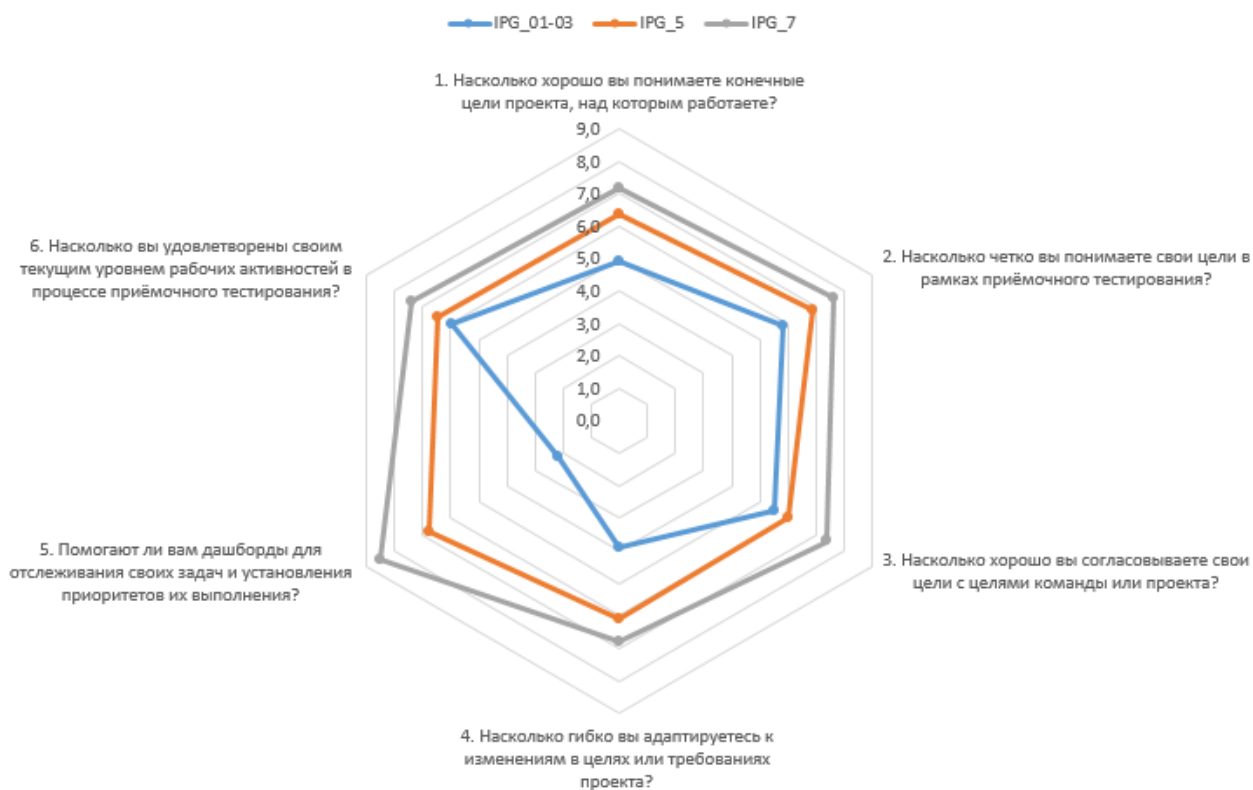


Рисунок 25 – Диаграмма среднего балла ответов по шкале на начало IPG_1-3, на IPG_5 и IPG_7

С момента введения в рабочий процесс разработанной модели дашбордов целеполагание сотрудников увеличивается. При этом осознаются не только цели приёмочного тестирования, но и общие проектные цели.

Использование новой модели дашбордов, по мнению сотрудников, ускоряет адаптацию к изменениям проекта. Также все участники рабочей группы отметили, что введённые дашборды помогают в определении своих задач и их приоритизации.

Поскольку для каждого нового опроса контуры линий расширяются от центра к периферии, то можно говорить, что уровень целеполагания и ориентирования в рабочем процессе повысился с момента введения дашбордов.

Выводы по третьей главе

В ходе исследования процесса разработки и оптимизации дашбордов для приёмочного (интеграционного) тестирования была разработана модель, учитывающая специфику данного процесса. Она включает адаптацию визуализации данных и создана с учетом потребностей пользователей, типов данных, ключевых метрик и показателей процесса тестирования. Анализ эффективности показал, что предложенная модель способствует повышению производительности в рабочей группе по проверке качества программного обеспечения, самоорганизации инженеров, а также улучшает целеполагание и ориентирование в рабочем процессе у всех участников. Применение разработанной модели из пяти дашбордов положительно повлияло на процессы приёмочного (интеграционного) тестирования.

Заключение

Выполненное исследование представлено следующими результатами:

- исследованы теоретические материалы об организации мониторинга в процессе тестирования программного обеспечения;
- подобраны необходимые метрики для мониторинга приёмочного (интеграционного) тестирования;
- описаны фильтры на основе выбранных метрик;
- создана модель дашборда, отражающая необходимые аспекты приёмочного процесса в актуальном состоянии;
- данная модель апробирована в периоды прохождения производственной практики и научно-исследовательской работы;
- рабочая группа по тестированию программного обеспечения обучена применению модели дашборда для получения рабочих задач и принятия решений о приоритизации работ;
- сделан вывод об эффективности предложенной модели;
- запущен процесс по распространению практики использования подобной модели на аналогичные процессы внутри компании ООО «НетКрэкер».

В рамках работы получилось создать такую модель дашбордов, которая позволила вывести рабочую команду проверки качества принимаемого программного обеспечения на уровень самоорганизации в части получения и приоритизации рабочих задач, но в то же время сохранила элементы для отчётов вышестоящим руководителям. Разработанная модель из 5 дашбордов содержит актуальную и максимально концентрированную информацию о процессе приёмки, интересную всем участникам рабочего процесса и управления.

Рабочая группа приёмочного (интеграционного) тестирования обучена использовать разработанную панель мониторинга в рабочем процессе. Все

исполнители хорошо ориентируются в назначаемых задачах благодаря разработанной модели дашбордов.

Применение оптимизированной модели позволило сократить сроки по объяснению нюансов организации приёмочного (интеграционного) тестирования. Применяемая модель повысила ориентирование в рабочем процессе.

Дополнительно следует отметить улучшение целеполагания в команде. Разработанные панели мониторинга позволили отследить как динамику проекта в целом, так и вклад каждого исполнителя в реализацию целей проекта, что опосредованно способствует снижению уровня профессионального выгорания.

Цель и задачи, поставленные в начале научно-исследовательской работы, достигнуты и реализованы в полной мере.

Эффективность применения разработанной модели дашбордов проверена и доказана. Гипотеза подтверждена.

Использование разработанного шаблона рекомендовано распространить на все процессы приёмки внутри компании ООО «НетКрэкер». Данную практику использования модели дашбордов можно применить для любых организаций, проводящих приёмочное (интеграционное) тестирование.

Развитие данной темы возможно продолжить в разработке и автоматизации отчётов о статусе проекта с использованием дашбордов с целью максимально возможного сокращения времени на написание таких отчётов.

Список используемых источников

1. Буянский С. Г., Трунцевский Ю. В. Корпоративное управление, комплаенс и риск-менеджмент: учеб. пособие для магистрантов. М. : Русайнс, 2016. 342 с.
2. Волкова В. Н., Васильев А. Ю., Ефремов А. А., Юрьев В. Н., Паклин Н. Б. Классификация информационных технологий // Журнал Прикладная информатика/Journal of applied informatics. 2015. Том 10, № 5(59). С. 124–141
3. Внутренние ресурсы ООО «НетКрэкер»
4. Голованова О. С. Дашборд – цифровой инструмент повышения эффективности управления человеческим капиталом // Инновационные технологии, экономика и менеджмент в промышленности : сборник научных статей по итогам второй международной научной конференции. 2021. – С. 93–95.
5. Горфинкель В. Я. Коммуникации и корпоративное управление : учеб. пособие М. : ЮНИТИ-ДАНА, 2015. 127 с.
6. Граничин О. Н. Информационные технологии в управлении : учебное пособие [Электронный ресурс] : Электронно-библиотечная система IPR BOOKS. URL: <http://www.iprbookshop.ru/89437.html> (дата обращения: 12.06.2022)
7. Гринберг А. С. Информационные технологии управления : учебное пособие для вузов [Электронный ресурс] : Электронно-библиотечная система IPR BOOKS. URL: <http://www.iprbookshop.ru/71234.html> (дата обращения: 10.06.2022)
8. Данышев Р. В. Автоматизация и оптимизация в управлении персоналом с использованием системы дашборд // Цифровая экономика: проблемы и перспективы развития : Сборник научных статей 3-й Межрегиональной научно-практической конференции. – Курск: Юго-Западный государственный университет, 2021. С. 120–124

9. Дунаевская О. В. Дашборд как основа самоорганизации команды во время приёмочных работ по проверке качества программного обеспечения // Материалы IX Международной научно-практической конференции (школы-семинара) молодых ученых «Прикладная математика и информатика: современные исследования в области естественных и технических наук». – Тольятти: ТГУ, 2023

10. Егорова Г. Н., Данилова Н. П. Аналитика данных с помощью построения дашбордов в MS Excel // Сборник материалов XVI Международно-практической конференции. – Чебоксары: Чебоксарский кооперативный институт (филиал) автономной некоммерческой образовательной организации высшего образования Центросоюза Российской Федерации "Российский университет кооперации", 2023. С. 172-176

11. Иванов Е. Сила дашбордов [Электронный ресурс] // URL: <https://software-testing.ru/library/testing/test-lab/3448-power-of-dashboards> (дата обращения 21.06.2023)

12. Короткий С. В. Корпоративное управление как элемент глобального инкорпорирования : монография. – Саратов : Вузовское образование, 2019. 124 с.

13. Крашенинникова А. А. Дашборд как инструмент успешного бизнеса // Вопросы экономики и финансов современное состояние актуальных проблем : Материалы Региональной научно-практической конференции, Калуга, 2019 – С. 79-84.

14. Криспин Л. Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд – М: И.Д. Вильямс, 2016. – 464 с

15. Крупина В. В. Визуализация данных средствами дашбордов // Информационные технологии и математическое моделирование в управлении сложными системами. 2019. № 2(3). С. 41–52.

16. Куликов С. С. Тестирование программного обеспечения. Базовый курс. Минск: Четыре четверти, 2017. – 312 с

17. Решение Федеральной службы по интеллектуальной собственности от 23 ноября 2018 г. по заявке N 1318193 О предоставлении правовой охраны «JIRA» [Электронный ресурс] // СПС ГАРАНТ: нормативно-правовая документация URL: <http://ivo.garant.ru/> (дата обращения 10.11.2023)
18. Савин Р. Тестирование Дот Ком, или Пособие по жесткому обращению с багами в интернетстартапах. М.: Дело, 2007. – 312с
19. Стандартный глоссарий терминов, используемых в тестировании программного обеспечения / International Software Testing Qualifications Board; под ред. Erik van Veenendaal, Александров А., 2014. 73 с.
20. Тюльпинова Н. В. Компьютерные и информационные технологии в науке и производстве : учебное пособие для магистров. Саратов : Вузовское образование, 2020. 268 с.
21. Уиттакер Дж. Как тестируют в Google. Санкт-Петербург: Питер, 2014. 320 с.
22. Успенский И. Всё о дашбордах: как они устроены, как компании их разрабатывают и как сделать их самому [Электронный ресурс] : Skillbox Media // URL: <https://skillbox.ru/media/> (дата обращения 18.06.2023)
23. Успенский И. Как создавать дашборды, которыми будут пользоваться [Электронный ресурс] // URL: <https://tproger.ru/articles/kak-sozdavat-dashbordy-kotorymi-budut-polzovatsja/> (дата обращения 21.06.2023)
24. Что такое KPI [Электронный ресурс] // URL: <https://developers.sber.ru/help/business-development/> (дата обращения: 11.10.2023)
25. A. Sarikaya, M. Correll, L. Bartram, M. Tory, and D. Fisher. What do we talk about when we talk about dashboards? IEEE Transactions on Visualization and Computer Graphics [Электронный ресурс] // URL: <https://arxiv.org/> (дата обращения 20.10.2023)
26. Alin Isac, Claudia Isac Informatics Issues Used in the Production Dashboard. [Электронный ресурс] // URL: <https://arxiv.org/abs/0905.4860> (дата обращения 20.10.2023)

27. Atlassian Support [Электронный ресурс] // URL: <https://confluence.atlassian.com/jira064/> (дата обращения 21.06.2023)
28. Azure DevOps. Создание панели мониторинга, не связанной с командой [Электронный ресурс] // URL: <https://azure.microsoft.com/ru-ru/updates/create-a-dashboard-without-a-team/> (дата обращения 21.06.2023)
29. Black R. Agile Testing Foundations, BCS Learning & Development Ltd: Swindon UK, 2017
30. GeekBrains. Дашборд: какие бывают и как создать [Электронный ресурс] // URL: <https://gb.ru/blog/dashbord/> (дата обращения 20.10.2023)
31. Habr. Хабр – ресурс для IT-специалистов [Электронный ресурс] // URL: <https://habr.com/ru/company/developersoft/blog/> (дата обращения 20.06.2023)
32. International Software Testing Qualifications Board Материалы Сертифицированный тестировщик в сфере гибких методологий: Программа подготовки ISTQB Базовый уровень – 2014 – 59с
33. International Software Testing Qualifications Board Сертифицированный тестировщик Программа обучения Базового уровня – Программа подготовки ISTQB Базовый уровень – 2018 – 96 с
34. ISO/IEC 20246: (2017) Software and systems engineering – Work product reviews [Электронный ресурс] // URL: <https://www.iso.org/> (дата обращения: 22.07.2023)
35. ISO/IEC 25010, (2011) Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models [Электронный ресурс] // URL: <https://www.iso.org/ru/standard/> (дата обращения: 15.09.2023)
36. ISO/IEC/IEEE 29119-1 (2013) Software and systems engineering - Software testing - Part 1: Concepts and definitions [Электронный ресурс] // URL: <https://www.iso.org/ru/standard/> (дата обращения: 18.06.2023)

37. ISO/IEC/IEEE 29119-2 (2013) Software and systems engineering - Software testing - Part 2: Test processes [Электронный ресурс] // URL: <https://www.iso.org/ru/standard/> (дата обращения: 15.09.2022)
38. ISO/IEC/IEEE 29119-3 (2021) Software and systems engineering - Software testing – Part 3: Test documentation [Электронный ресурс] // URL: <https://www.iso.org/ru/standard/> (дата обращения: 10.07.2023)
39. John Jarske, and Jorge Rady, and Lucia V. L. Filgueiras, and Leandro M. Velloso, and Tania L. Santos. Modeling the Dashboard Provenance, 2023 [Электронный ресурс] // URL: <https://arxiv.org/pdf/2308.06788.pdf> (дата обращения 18.10.2023)
40. J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. IEEE Transactions on Visualization and Computer Graphics, 13(6), 2007. doi: 10.1109/TVCG.2007.705942 [Электронный ресурс] // URL: <https://arxiv.org/> (дата обращения 20.10.2023)
41. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Towards a general-purpose query language for visualization recommendation. [Электронный ресурс] // URL: <https://arxiv.org/> (дата обращения 15.09.2023)
42. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. [Электронный ресурс] // URL: <https://arxiv.org/> (дата обращения 20.10.2023)
43. Kramer A., Legard B. Model-Based Testing Essentials: Guide to the ISTQB Certified, 2016
44. Netcracker Technology //статья из Википедия. Свободная энциклопедия [Электронный ресурс] // URL: https://ru.wikipedia.org/wiki/Netcracker_Technology (дата обращения 10.01.2022)
45. Skillbox. Платформа профессионального и личного развития [Электронный ресурс] // URL: <https://skillbox.ru/> (дата обращения 19.06.2023)

46. Stephen Few Information Dashboard Design: The Effective Visual [Электронный ресурс] // URL: <https://arxiv.org/> (дата обращения 20.10.2023)
47. Thomas Hamilton What is System Integration Testing (SIT) Example [Электронный ресурс] // URL: <https://www.guru99.com/system-integration-testing.html> (дата обращения 18.10.2023)
48. Tilo Linz, «Testing in Scrum: A Guide for Software Quality Assurance in the Agile World», Rocky Nook, 2014.

Приложение А
Вопросы для анкетирования инженеров по проверке качества ПО
Анкета

1. Насколько хорошо вы понимаете конечные цели проекта, над которым работаете? (0 – совсем не понимаю, 10 – очень четко понимаю)



2. Насколько четко вы понимаете свои цели в рамках приёмочного тестирования? (0 – совсем не понимаю, 10 – очень четко понимаю)



3. Насколько хорошо вы согласовываете свои цели с целями команды или проекта? (0 – совсем не согласовываю, 10 – очень хорошо согласовываю)



4. Насколько гибко вы адаптируетесь к изменениям в целях или требованиях проекта? (0 – не адаптируюсь, 10 – очень гибко адаптируюсь)



5. Помогают ли вам дашборды для отслеживания своих задач и установления приоритетов их выполнения? (0 – неэффективно, 10 – очень эффективно)



6. Насколько вы удовлетворены своим текущим уровнем рабочих активностей в процессе приёмочного тестирования? (0 – совсем не удовлетворен, 10 – очень удовлетворен)



Приложение Б
Бизнес-план по оптимизации процесса приёмочного (интеграционного) тестирования

Таблица Б.1 – Бизнес план по реализации ценностного предложения

Элементы	Описание
Цели	<p><i>Краткосрочные цели:</i> Сократить временные затраты руководителей рабочих групп на административные работы. Повысить уровень самоорганизации сотрудников рабочих групп. Разработать модель дашборда</p> <p><i>Долгосрочные (стратегические) цели:</i> Сократить время приёмочного (интеграционного) тестирования с 3 месяцев до 3-4 недель.</p>
Миссия	<p><i>«Долой профессиональное выгорание! Будь мультипотенциалом администрируй и тестируй!»</i></p> <p>Комментарий к миссии: Обычно руководителями команд тестирования становятся опытные тестировщики. Значимость работы таких руководителей не должна сводиться только к управлению. Чтобы грамотно управлять проектом, следует хоть немного времени затрачивать на то, чтобы «пощупать» его изнутри, быть вовлечённым в рабочий процесс, не потерять квалификацию в тестировании.</p>
Компетенции	<p>Ключевой вид деятельности: тестирование программного обеспечения.</p> <p>Компетенции по созданию ценностного предложения: В итеративном процессе (на примере интеграционного тестирования) очень важно, чтобы руководитель группы сохранял свои профессиональные навыки (тестирования), а также чётко руководил процессом (приёмочных работ) и своевременно составлял отчёты. Для реализации целей НИР необходимо найти таких руководителей групп, которые смогут легко ориентироваться в многозадачности, будут быстро переключаться, а также будут готовы выполнять не только работу по управлению, но и работу в рамках непосредственно тестирования.</p>
Планы	<p>Маркетинговый: Разработать модель дашборда. Обучить сотрудников применять уже созданные дашборды в своей работе. За счёт экономии времени на управление рабочей группой, руководитель частично посвящает свой рабочий день непосредственному тестированию.</p> <p>Привлечения персонала: Набрать рабочую группу из 10 человек в соотношении: опытных сотрудников – 60%, новичков – 40%</p> <p>Материально-технического обеспечения (ресурсов): Каждому сотруднику рабочей группы предоставить персональный аккаунт для работы в сервисах: JIRA, TMS, Confluence, Webex, Zoom, Outlook. Каждому сотруднику рабочей группы предоставить ноутбук с дополнительным монитором.</p> <p>Производственный: <i>Помещение</i> – допустимо работать из дома с обязательным посещением офиса 2 раза в неделю.</p>

Элементы	Описание
	<p><i>Время работы</i> – 8 часовой рабочий день, 5 дней в неделю.</p> <p><i>Производственный процесс</i> – ООО «НетКрэкер» разрабатывает программное обеспечение для внедрения и сопровождения систем поддержки бизнеса (Business Support System) и систем операционной поддержки (Operation Support System). Функционал такого программного обеспечения зависит от конфигурации, запрошенной заказчиком. Конфигурация выстраивается из набора компонент программного обеспечения. Каждая компонента создаётся в отдельной команде, в которую входят проектные менеджеры, бизнес-аналитики, разработчики и инженеры по контролю качества, а затем выдаётся для приёмки в релиз. Релизы программного обеспечения формируются 4 раза в год. Команда релиза занимается приёмкой разработанных компонентов или обновлений к ним и последующим тестированием всех возможных версий релиза.</p> <p><i>Постоянные расходы</i> – оплата лицензий используемого ПО, выплаты заработной платы сотрудникам.</p>
Издержки	<p>Ежегодная инфляция – 10%;</p> <p>НДФЛ – 13%;</p> <p>НДС – 18%</p> <p>Социальные отчисления – 34,2%</p>
Риски	<p>Риски, связанные с рабочими кадрами: риски отсутствия у сотрудников необходимых профессионально значимых качеств; риски отсутствия у сотрудников необходимых личностных качеств; риски неадекватной мотивации сотрудников; риски нелояльности.</p> <p>Риски санкций: ООО «НетКрэкер» является российским филиалом американской компании «Netcracker» и входит в японскую корпорацию NEC. Российские компании подвержены санкционному риску на фоне геополитической напряженности.</p> <p>План реагирования и минимизации рисков:</p> <p>На риски санкций сотрудники компании повлиять не могут. Однако мы можем минимизировать риски, связанные с движением кадров: своевременно обучать членов рабочей группы, отслеживать их уровень удовлетворённости своими задачами, назначать дополнительные вознаграждения в случае высокого качества работ.</p>

Приложение В

Преимущества и недостатки «дашборда для руководителей» и «дашборда для всех участников рабочего процесса»

Таблица В.1 – Плюсы и минусы в сравнении дашборда для руководителей и дашборда для всех участников рабочего процесса

Критерии	Дашборд для руководителей		Дашборд для всех участников	
	Плюсы	Минусы	Плюсы	Минусы
<i>Визуализация данных</i>	да		да	Сложнее настраивать, т.к. требуется отразить большее количество метрик
<i>Централизация информации</i>		нет	да	
<i>Удобство и краткость</i>	да	Недостаток детальной информации; Неэффективность в отношении специфических требований	не всегда; детализация на высоком уровне	
<i>Способность быть настроенным индивидуально для каждого</i>	да	Ограниченная гибкость; Руководители могут иметь ограниченный доступ к данным, что может затруднить получение полной картины ситуации		Нет; Недостаточная гибкость: пользователи не могут адаптировать интерфейс дашборда к своим индивидуальным предпочтениям

Продолжение Приложения В

продолжение таблицы В.1

Критерии	Дашборд для руководителей		Дашборд для всех участников	
	Плюсы	Минусы	Плюсы	Минусы
Улучшение коммуникации	Коммуникация происходит между проектными менеджерами и руководителями групп	Исполнители тестирования могут выпадать из коммуникаций	Коммуникация происходит между всеми участниками рабочего процесса	
Мониторинг и быстрые реакции	да	Мониторинг проводит проектный менеджер	да; мониторинг можно делегировать руководителям рабочих групп	
Принятие решений	Руководители рабочих групп могут повлиять на принятие решения	Принятие решений лежит на проектном менеджере; Зависимость от данных в реальном времени	Каждый участник рабочего процесса может повлиять на принимаемое решение	Зависимость от данных в реальном времени
Мониторинг производительности исполнителей		Возможно учитывать только по общей производительности рабочей группы	Возможно учитывать только по каждому сотруднику	
Угроза безопасности		есть		есть
Время на создание отчётов		Затрачивается много времени, т.к. часть информации нужно постоянно детализировать из других источников	Время сокращается за счёт большого количества детализаций	

Критерии	Дашборд для руководителей		Дашборд для всех участников	
	Плюсы	Минусы	Плюсы	Минусы
<i>Нужно обучать рабочую группу использованию дашборда</i>	нет			Да
<i>Передача знаний по процессу приёмки</i>		Передача знаний возможна только вышестоящему лицу (от руководителя группы – к проектному менеджеру)	Передать знания можно любому участнику рабочей группы	
<i>Ориентирование в рабочем процессе</i>		Ориентирование осуществляет руководитель группы как для проектного менеджера, так и для исполнителей тестирования	Ориентирование осуществляется каждым сотрудником самостоятельно	Проектный менеджер нуждается в отдельном ориентировании
<i>Самоорганизация рабочей группы</i>		Рабочая группа нуждается в постоянном контроле и поддержке	Члены рабочей группы способны к самоорганизации в части получения заданий и их приоритизации	
<i>Целеполагание внутри команды</i>		На среднем уровне	На высоком уровне	

Приложение Г
Уровни тестирования по степени детализации приложения

Таблица Г.1 – Описание уровней тестирования по степени детализации приложения

Уровни тестирования	Цели тестирования	Базис тестирования	Объект тестирования	Типичные дефекты
Компонентное тестирование	Снижение риска. Проверка, соответствует ли функциональное и нефункциональное поведение компонентов установленным проектным требованиям. Укрепление уверенности в качестве компонента. Обнаружение дефектов в компоненте. Предотвращение пропуска дефектов на более высокие уровни тестирования.	Детальный дизайн. Код. Модель данных. Спецификации компонента.	Компоненты, модули. Код и структуры данных. Классы. Модули базы данных (БД).	Неправильная работа функциональности (например, не так, как описано в спецификации). Проблемы с потоками данных. Неправильные код и логика работы компонента или отдельных его частей.

Продолжение Приложения Г

продолжение таблицы Г.1

Уровни тестирования	Цели тестирования	Базис тестирования	Объект тестирования	Типичные дефекты
Интеграционное тестирование	<p>Снижение риска появления дефектов на более поздних этапах. Проверка, соответствует ли функциональное и нефункциональное поведение интерфейсов установленным проектным требованиям. Повышение уверенности в качестве интерфейсов. Обнаружение дефектов (которые могут быть в самих интерфейсах или внутри компонентов/систем). Предотвращение пропуска дефектов на более высокие уровни тестирования.</p>	<p>Дизайн продукта и системы. Диаграммы последовательности. Спецификации интерфейса и протокола связи. Сценарии использования системы. Архитектура на уровне компонентов или системы. Рабочие процессы. Спецификации, описывающие внешние интерфейсы.</p>	<p>Подсистемы. Базы данных. Инфраструктура. Интерфейсы. Программные интерфейсы приложения (API). Микросервисы.</p>	<p>Некорректные данные, отсутствующие данные или неправильная кодировка данных. Неверная последовательность или временные характеристики обращения к интерфейсам. Несовместимость интерфейсов. Сбои связи между компонентами/системами. Необработанные или неправильно обработанные сбои связи между компонентами/системами. Неправильные предположения о назначении, единицах или границах данных, передаваемых между компонентами/системами. Несогласованные структуры сообщений между системами. Несоблюдение обязательных правил безопасности.</p>

Продолжение Приложения Г

продолжение таблицы Г.1

Уровни тестирования	Цели тестирования	Базис тестирования	Объект тестирования	Типичные дефекты
Системное тестирование	Снижение риска. Проверка, соответствует ли функциональное и нефункциональное поведение системы установленным проектным требованиям, дизайну и спецификациям. Проверка, что система реализована полностью и будет работать, как ожидалось. Повышение уверенности в качестве системы в целом. Обнаружение дефектов. Предотвращение попадания дефектов на более высокие уровни тестирования или в среду эксплуатации.	Системные требования и требования к продукту (функциональные и нефункциональные). Отчеты об анализе рисков. Сценарии использования. Бизнес-потребности и пользовательские истории. Модели поведения системы. Диаграммы состояний. Системные и пользовательские руководства.	Приложения. Аппаратные/программные системы. Тестируемая система. Операционные системы. Конфигурация системы и конфигурация данных.	Некорректные вычисления. Некорректное или неожиданное функциональное или нефункциональное поведение системы. Некорректное управление и/или передача данных внутри системы. Невозможность правильно и полностью выполнить функциональные задачи конечными пользователями. Неспособность системы работать правильно в среде эксплуатации. Неспособность системы работать так, как описано в системных и пользовательских руководствах.