**Предложения для улучшения функционала TestIT**

* 1. Возможность создавать группы из пользователей

*Если внутри коллектива есть распределение крупных задач, и один сотрудник может вести несколько разных задач, то была бы удобна возможность назначать тест-планы на группу.*

* 1. Возможность добавлять разное количество объектов из выборных параметров.

*Реализация pairwise уже существующими средствами. Сейчас все поля должны быть заполнены обязательно.*

* 1. Возможность преобразовать:
* тест-кейс в чек-лист
* тест-кейс на общий шаг
* чек-лист в общий шаг
* общий шаг в тест-кейс
* общий шаг в чек-лист

*Сейчас есть только возможность преобразовать чек-лист в тест-кейс.*

* 1. Возможность в рамках одного кейса добавлять версии функционала.

*Если вышла новая версия ПО и в его рамках функционал сохранился, но изменился. Есть необходимость не изменять кейс, а иметь возможность отслеживать изменения в реализации по версиям.*

* 1. К действиям с кейсом добавить возможности
* отправить на автоматизацию
* отправить на проверку
* отправить на редакцию/исправление/актуализацию
* отправить на описание *(кейс создан, присвоено название, но не заполнен)*
  1. В профиле сотрудника в “Задачах” так же, как и тест-план добавить отображение задач из п.5.

*Отображаются и по выполнению исчезают кейсы, которые были назначены на автоматизацию, проверку, редакцию и описание.*

* 1. К действиям в кейсе (или шаге), или как отдельный вид кейса добавить возможность формировать проверки
* с помощью pairwise (в вариантах “все сочетания” и “минимальное количество сочетаний”)
* с помощью матрицы

*В проверках часто необходимо учитывать сочетание 2х и более параметров. Описывать каждое сочетание в отдельной проверке – трудоёмко и неудобно. Выборные параметры не всегда уместно использовать, так как количество параметров может отличаться. Кроме того, существует функционал, в котором сочетание А+В≠В+А, т.е. каждое сочетание необходимо проверять отдельно.*

* 1. Отобразить к характеристикам кейса кроме приоритета по умолчанию ещё уровень проверок.

*Сейчас это можно реализовать через атрибут. Но у ошибок ПО существуют соответственные характеристики: приоритет и серьёзность (критичность). Логично, если и у кейсов, которые нацелены на выявление этих проверок, будут такие же характеристики. Серьёзность подразумевает критичность функционала. Например: старт программы – 1ый уровень, выполнение функции – 2ой уровень, валидность полей – 3ий уровень. Уровни можно использовать или не использовать, их количество пользователь может варьировать и редактировать их названия.*

* 1. Собрать в одно место параметры кейсов и добавить возможность работать с ними.
* теги: отображение используемых/неиспользуемых, статистика использования, возможность просмотра проектов/кейсов, в которых они используются
* выборные параметры: отображение используемых/неиспользуемых, создание/удаление/редактирование, статистика использования, возможность просмотра проектов/кейсов, в которых они используются
* уровни проверок (п.8): создание/удаление/редактирование уровней проверок, возможность выделить в отдельный элемент структуры кейсы по уровню.
  1. Добавить возможность добавления прямых ссылок на клиентские ошибки в redmine (с возможностью прикреплять несколько ошибок к одному кейсу и одну ошибку к разным кейсам), на статьи в confluence.
  2. Расширить возможности работы с изображениями в кейсах:
* вид отображения (плитка, крупные значки, таблица, обычные значки)
* возможность мультивыделения и удаления добавленных изображений
* пометка для изображений, которые актуальны в кейсе (или отсутствуют в нём)
* удалять изображения вместе с удаляемыми шагами
  1. Возможность собрать единую матрицу по проекту или направлению.

Варианты отображения включают в себя взятую за основу структуру. В текущей ситуации предлагаем для реализации 2 варианта: таблица и графы.

* Таблица – стандартное отображение для любой матрицы. За основу берутся 2 параметра. Один из них уже существует – древовидная структура (ось Y). Для оси Х можно выбрать любой пользовательский параметр, или использовать уровень проверок (п. 8).
* Графы для отображения более сложные (активно используются в Obsidian). В их основу можно взять уже существующие связи через общие шаги и ссылки между кейсами. Или использовать теги, а чтобы не перегружать связи – использовать попарное совпадение.
  1. Возможность отображать структуру проекта не вертикально, а горизонтально, с возможностью масштабировать выбранный уровень/ветку или элемент структуры и перемещать их для удобства восприятия
  2. Структуру проекта можно отображать, учитывая только номера кейсов или только названия
  3. На сформированную матрицу можно наложить отображение по любому параметру кейса, в том числе и авторскому атрибуту. Но особенно важными можно считать матрицу тестирования (на release candidate) и матрицу клиентских ошибок (п.10)

*Любой выбранный для отображения в матрицу параметр будет визуально отображать структурные элементы (ветки, уровни) или функционал (кейсы), которые не учтены этим параметром или наоборот – перегружены.*

* 1. Возможность создания mind-map

*В основу mind-map можно положить уже существующую структуру. Mind-map необходима для понимания связей функционала внутри проекта, узловых моментов и т.д. Либо сделать возможной интеграцию, например, с draw.io.*

* 1. Возможность использовать номера удалённых кейсов

*При долгой жизни проекта номера быстро становятся 5тизначными, что усложняет их восприятие и отображение.*

* 1. Возможность создавать тест-планы без полного раскрытия структуры: выбирать по тегу, ветку и т.д.
  2. Импорт из Excel/Google уже существующих матриц
  3. Возможность переноса кейса из одного проекта в другой
  4. Возможность добавить общий шаг в тест-план

*Часто общий шаг является предусловием для более сложного функционала*