

- выделение контуров и распознавания объектов.

После анализа гистограммы можно будет получить информацию об уровне яркости и выявить необходимость его корректировки. Это позволит улучшить качество изображений, полученных в результате фотографирования в условиях недостаточного освещения, и изображений, полученных в условиях избыточно экспозиции.

Различные двумерные операторы помогут идентифицировать проблемы размытия изображения и подготовить изображение для распознавания объектов. В дальнейшем размытие изображения может быть исправлено фильтром Винера, фильтром Люси-Ричардсона и другими способами.

Распознавание объектов в свою очередь поможет выделить лица, пейзаж, силуэты людей и другие объекты, часто встречающиеся на изображениях. После этого будет проведено выравнивание горизонта, исправление «красных глаз», улучшение цвета кожи.

Преимущества предложенного способа автоматизации редактирования изображений:

- выполняет монотонные и типовые действия за художника, что позволяет экономить время;
- детектирование бракованных изображений, не подлежащих восстановлению, из большого множества изображений;
- выполнение трудоёмких действий.

Недостатком является то, что интеллектуальные алгоритмы ничего не знают о замысле художника. Некоторые дефекты изображения: красные глаза, нестандартный наклон горизонта, размытие – могут быть внесены художником сознательно, но алгоритм всё равно попытается их исправить. При высоких разрешениях изображения процесс интеллектуальной обработки может занимать большие промежутки времени и является малоприменимым для обработки больших массивов изображений. Таким образом, интеллектуальное редактирование изображений подходит только для типовых изображений небольших разрешений и не подходит для редактирования художественных снимков.

Список использованных источников:

1. Andrew Faulkner, Conrad Chavez Adobe Photoshop CC Classroom in a Book / Adobe Press – 2018, - 559с.
2. Olivier Lacarne, Karine Delvare The Book of GIMP / No Scratch Press – San Francisco 2013, - 679с.

SCRUM: ОБУЧЕНИЕ И РЕАЛЬНОСТЬ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Чехович Е.С.

Искра Н.А. – ст. преподаватель

На сегодняшний день большинство ведущих IT-компаний используют в своей работе Agile-методологию, популярнейшим фреймворком в рамках которой является Scrum [1]. Белорусские компании также внедряют данный фреймворк в рамках разработки проектов.

Согласно отчёту о популярности Agile [2], 28% европейских компаний используют данную методологию, а 58% из них работают с фреймворком Scrum. Однако, даже те, кто работают в этих компаниях, не всегда в полной степени владеют данным фреймворком.

Такая ситуация приводит к увеличению количества вакансий, где одним из требований к соискателю является владение основами Scrum. К сожалению, мало в каких университетах в программе предусмотрено изучение фреймворка, а если его и изучают, то только “на бумаге”, т.е. практических навыков специалист не получает.

Проблема. Соответствие специалистов требованиям работодателей.

Пути решения. Обучать. Но подходов к обучению много.

На практике в IT-компаниях фреймворк Scrum преподаётся в виде 1-2 дневного семинара, в рамках которого читается краткий курс теории и проводится несколько “игр в лего”. Но самое важное в изучении Scrum “адаптировать” процесс и “адаптироваться” к нему, т.е. попробовать.

Вот уже 3 года на кафедре ЭВМ в рамках дисциплины “Технологии разработки и тестирования ПО” [3] практикуется изучение фреймворка Scrum в “реальном” процессе разработки проектов.

Однако полностью воссоздать рабочий процесс невозможно. Есть как положительные, так и отрицательные отличия обучающего процесса от рабочего.

Положительные отличия:

1. Если в процессе обучения команда допустит ошибки, влияющие на качество или скорость разработки проекта, ничего страшного не произойдёт. Но в реальном процессе такие ошибки недопустимы и могут привести к провалу проекта.
2. Во время обучения команда может выбрать новую или интересную для себя технологию и потратить время на её изучение. В реальности такой подход к выбору технологий недопустим и, чаще всего, команды годами работают в рамках тех технологий, которые им хорошо известны.
3. В процессе обучения студент, как правило, получает оценку своей работы и удовлетворительной оценкой может считаться довольно низкой.

Отрицательные отличия:

1. Вознаграждение в обучающем процессе или не предусматривается или предполагается после завершения проекта. В реальном процессе команда работает не на перспективу, а получает быстрый отклик на свою работу.
2. Невозможность сэмулировать процесс полностью. Ежедневный митинг невозможен в принципе, поскольку занятия проходят раз в неделю. Также невозможно воссоздать полный рабочий день, поскольку учебной программой предусмотрено времени гораздо меньше.

В общем случае можно уменьшить влияние отрицательных отличий. Способы уменьшения были выявлены опытным путём в процессе разработки проектов.

Одним из способов уменьшения влияния является привязка графика работы к расписанию занятий. Таким образом нивелируется сложность воссоздания рабочего процесса, а у команды вырабатывается ритм работы и нет соблазна отложить разработку проекта на конец спринта.

Снизить влияние отрицательных отличий позволяет использование средств удалённого общения и выбор проектов небольшого размера или проектов, с которыми команды уже работали.

Таким образом, используя все положительные отличия и уменьшая влияние отрицательных, возможно изучать Scrum в “реальном” процессе разработки проектов. В итоге студенты получают практические навыки работы в рамках данного фреймворка, что в будущем поможет им быстрее влиться в рабочий процесс.

Список использованных источников:

1. ScrumAlliance: The Scrum Guide [Электронный ресурс]. – Режим доступа: <https://www.scrumalliance.org/learn-about-scrum/the-scrum-guide> – Заглавие с экрана. – (Дата обращения: 10.04.2018).
2. 11th Annual State of Agile Report [Электронный ресурс]. – Режим доступа: <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2> – Заглавие с экрана. – (Дата обращения: 12.04.2018).
3. Искра, Н. А. Технологии разработки и тестирования программного обеспечения [Электронный ресурс] / Н. А. Искра // Учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-40 02 01 Вычислительные машины, системы и сети – 2015. – Режим доступа: <https://libeldoc.bsuir.by/handle/123456789/30530>. – (Дата обращения: 12.04.2018).

СИСТЕМА КОНТРОЛЯ СОСТОЯНИЯ КОМПОНЕНТОВ ЛОКАЛЬНОЙ СЕТИ

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шантарович В.Д.

Ганжа В.А. – к. ф.-мат. н, доцент

Концепция Internet of Things, как множество физических объектов (вещей), взаимодействующих с друг другом и средой, является нашим будущим и от него не убежать. Вещи вокруг нас становятся умнее с каждым днём и многие используемые задачи должны быть автоматизированы. Так как наша зависимость от IoT растёт с каждым днём, мы должны быть уверены, что такие устройства работают надёжно, безопасно и выполняют задачи, возложенные на них. Реализация такой системы контроля требует безопасную операционную систему для коммуникации с вещами, так как они могут пересылать персональную информацию, и, одновременно, эффективную, так как аппаратные ресурсы IoT скудны. Вот где Zabbix вступает в игру. Zabbix – высоко интегрированное решение мониторинга сети, которое предлагает множество функций в одном пакете.

Zabbix – это программное обеспечение для мониторинга многочисленных параметров сети, жизнеспособности и целостности серверов. Zabbix использует гибкий механизм оповещений, что позволяет пользователям конфигурировать уведомления основанные на e-mail практически для любого события. Это позволяет быстро реагировать на проблемы с серверами. Zabbix предлагает отличные функции отчетности и визуализации данных основанные на данных истории. Это делает Zabbix идеальным для планирования мощности[1].

Zabbix предлагает крошечного, скромного к ресурсам, агента для таких устройств: датчики в доме, торговый аппарат, электронное устройство и так далее. Данный агент собирает данные о производительности, доступности, статусе данных и других, различных метриках, передавая их другим устройствам или в облако. Создаётся истинная синергия между вещами IoT.

Zabbix состоит из нескольких основных программных компонентов.

Zabbix-сервер является основным компонентом, которому агенты сообщают информацию и статистику о доступности и целостности. Сервер является главным хранилищем, в котором хранятся все данные конфигурации, статистики, а также оперативные данные.

Как таковая вся информация о конфигурации, а также данные собранные Zabbix, хранятся в базе данных.

Для легкого доступа к Zabbix из любого места и с любой платформы, поставляется интерфейс на основе Веб. Интерфейс является частью Zabbix сервера и обычно (но не обязательно) работает на том же самой физической машине, что и сервер.