Задачи и методы моделирования и анализа программных продуктов

Теоретическое занятие

Специальность: 09.02.07 Информационные системы и программирование

МДК.03.01 Моделирование и анализ программного обеспечения

Тема 3.1.1 Задачи и методы моделирования и анализа программных продуктов

Преподаватель: Процюк М.П.

2022

Методы организации работы в команде разработчиков. Системы контроля версий

Подготовка специалистов для действий в составе команды является одной из современных задач подготовки кадров высшей квалификации. В обычной практике, разработка дипломных и курсовых проектов осуществляется выпускником и студентом единолично, под руководством научного руководителя, при этом не формируются навыки работы в команде, не приобретается опыт работы с инструментальными средствами коллективной разработки проектов, не формируются навыки руководства проектами. На производстве же молодой специалист сразу получает задание в составе группы. Особенно ярко это проявляется при реализации крупных проектов в области создания программного обеспечения. Отсутствие навыков работы в команде сказывается на производительности молодого сотрудника и его адаптации в рабочем процессе. Важным фактором является целостность ведения всего проекта и контроля над его реализацией. Как часто бывает в реальной работе, увольнение одного из разработчиков приводит к потере целых фрагментов проекта, так как при работе в простой студии, всё находится в компьютере программиста.

Другой не менее важный вопрос - наличие методики реализации проекта, очень важно для любого дела. Привить все эти навыки очень важно со студенческой скамьи.

Современные средства разработки программного обеспечения частично позволяют решить данные проблемы.

Итак, при использовании Visual Studio Team System решаются следующие задачи:

- Повышение предсказуемости успеха проекта.
- Рост производительности труда команды разработчиков за счет понижения сложности процессов проектирования и реализации современных сервисно-ориентированных решений.

- Обеспечение сотрудничества команды путем интеграции коммуникационных и прочих средств, используемых ее членами.
- Расширение возможностей командной работы за счет обеспечения удаленным пользователям доступа к надежному, защищенному и масштабируемому окружению.
- Предоставление команде разработчиков гибкого и расширяемого коммуникационного решения с настраиваемыми сервисами.

Ключевыми членами проектной команды являются: руководитель проекта, архитектор, разработчик и тестировщик. Каждую из этих ролей может исполнять как один человек, так и несколько. Второстепенные участники проекта, являющиеся профессионалами в области информационных технологии, также по достоинству оценят использование Team System, так как эта система облегчит им взаимодействие с остальными членами команды.

Научному руководителю курсовых и дипломных проектов, выполняемых в составе команды, необходимо также решать следующие задачи:

- Психологическая совместимость.
- Методы подбора групп.
- Сроки введения группового проектирования, первоначальная подготовка.
- Распределение ролей, ротация ролей.
- Другой не менее важный вопрос - наличие методики реализации проекта, что очень важно для любого дела.

Основные постулаты и принципы работы в команде

Сегодня основным ресурсом эффективного ведения хозяйственной деятельности предприятия является корпус специалистов. Резко возрастает роль личностей. От их квалификации, деловой активности, умения взаимодействовать друг с другом и достигать результата зависят перспективы развития предприятия. Одним из наиболее востребованных качеств, наряду с профессионализмом, является способность работать в команде. Ключевым фактором эффективной работы команды является способность каждого ее члена «работать на результат».

Однако на практике иногда случается так, что психологический аспект смещается с результатов деятельности на межличностную конкуренцию, проявляющуюся в скрытой или явной конфронтации.

В вопросе формирования команды важным элементом является «численность». Пять - подходящее количество людей, двенадцать - слишком много. "7 ± 2" является магическим числом и идеальной численностью команды.

В случае если в Вашем коллективе 12 и более человек, рано или поздно естественным образом он поделится на «неформальные команды по интересам», представляя в общем виде разрозненную и неконсолидированную психологически массу. Этот процесс может быть предупрежден руководителем: нужно осознанно объединить сотрудников в команды вокруг определенных профессиональных целей. Когда одна цель будет достигнута, важно вновь и вновь объединять сотрудников в команды, создавая и поддерживая, таким образом, командный дух в коллективе и поддерживая, повышая мотивационный уровень сотрудников.

Состав команды является самой важной предпосылкой высокой производительности компании. При подборе персонала нужно учитывать три фактора.

Фактор первый: профессиональная квалификация. Профессиональные требования зависят от конкретных задач той или иной команды, поэтому до начала формирования команды следует составить список требований, связанных с конкретной задачей.

Фактор второй: личностные характеристики.

Фактор третий: способность работать в команде. Это может быть как предпосылкой, так и результатом развития команды. Без базовой способности сотрудничества с другими членами команды, ориентируясь на результат, ничего не получится. Следовательно, при формировании команды требование к способности работать в ней выдвигается в первую очередь. Необходимо определить хочет ли и может ли кандидат работать в команде, и в какой степени.

Классификация систем контроля версий:

1) Централизованные/распределённые - в централизованных системах контроля версий вся работа производится с центральным репозиторием, в распределённых - у каждого разработчика есть локальная копия репозитория.

2) Блокирующие/не блокирующие - блокирующие системы контроля версий позволяют наложить запрет на изменение файла, пока один из разработчиков работает над ним, в неблокирующий один файл может одновременно изменяться несколькими разработчиками.

3) Для текстовых данных/для бинарных данных - для VCS для текстовых данных очень важна поддержка слияния изменений, для VCS с бинарными данными важна возможность блокировки.

Возможности систем контроля версий:

1) Поддержка хранения файлов в репозитории.

2) Поддержка истории версий файлов в репозитории.

3) Нахождение конфликтов при изменении исходного кода и обеспечение синхронизации при работе в многопользовательской среде разработки.

4) Отслеживание авторов изменений.

Обзор существующих инструментальных средств для командной работы

Цели, задачи, этапы и объекты ревьюирования. Планирование ревьюирования

Ревьюирование (инспекция кода) (Code review)[1]– систематический и периодический анализ программного кода, направленный на поиск необнаруженных на ранних стадиях разработки программного продукта ошибок, а также, на выявление некачественных архитектурных решений и критических мест в программе.

Задачи и цели проведения формальных инспекций

Не всегда возможна разработка автоматических или хотя бы четко формализованных ручных тестов для проверки функциональности программной системы. В некоторых случаях выполнение тестируемого программного кода

[1] Шуваев А.В. Программная инженерия. – Ставрополь: Ставропольский государственный аграрный университет, 2021.

невозможно в условиях, создаваемых тестовым окружением. Такая ситуация возможна во встроенных системах, если программный код предназначен для обработки исключительных ситуаций, создаваемых только на реальном оборудование.[2]

В тех случаях, когда верифицируется не программный код, а проектная документация на систему, которую нельзя "выполнить" или создать для нее отдельные тестовые примеры, также обычно прибегают к методу экспертных исследований программного кода или документации на корректность или непротиворечивость.

Такие экспертные исследования называют инспекциями или просмотрами. Существует два типа инспекций - неформальные и формальные.

При неформальной инспекции автор некоторого документа или части программной системы передает его эксперту, а тот, ознакомившись с документом, передает автору список замечаний, которые тот исправляет. Сам факт проведения инспекции и замечания нигде отдельно не сохраняются, состояние исправлений по замечаниям также нигде не отслеживается.

[2] Шуваев А.В. Программная инженерия. – Ставрополь: Ставропольский государственный аграрный университет, 2021.

Список использованной литературы

1. Гагарина Л.Г. Технология разработки программного обеспечения: учебное пособие. – М.: ИНФРА-М, 2022. – 400 с.

2. Марчуков А.В., Савельев А.О. Работа в Microsoft Visual Studio. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 384 с.

3. Рудаков А.В. Технология разработки программных продуктов: учебник для СПО. – М.: Академия, 2018. – 208 с.

4. Шуваев А.В. Программная инженерия. – Ставрополь: Ставропольский государственный аграрный университет, 2021. – 84 с.