

Конкретные модели эвристического обучения: эвристический диалог; мозговая атака; синектика; различные эвристические приемы решения задач (инверсия, специализация, редукция, аналогия, метод эвристических вопросов, агглютинация, эмпатия); теория решения изобретательских задач (ТРИЗ); проектное обучение; педагогические («французские») мастерские; кейс-метод (анализ конкретных ситуаций); технология развития критического мышления; моделирование и интерпретация объекта познания; бинарные оппозиции; деловые и организационно-деятельностные игры. Применение указанных моделей эвристического обучения создает возможность перехода от информационной парадигмы образования к смысловой парадигме как основе для выбора обучающимся и вариативного воплощения субъективно значимых достижений в творческие образы, их презентации в доступном и референтном окружении.

#### ЛИТЕРАТУРА

1. Кашлев, С.С. Современные технологии педагогического процесса: пособие для педагогов / С.С. Кашлев. – Минск: Университетское, 2000. – 95 с.
2. Хуторской, А.В. Развитие одаренности школьников: методика продуктивного обучения: пособие для учителя / А.В. Хуторской. – М.: Гуманит. изд. центр «Владос», 2000. – 320 с.

УДК 621.762.4

Шнитко А.В., Мелихов В.А.

#### **ПРАКТИКА ПАРНОГО ПРОГРАММИРОВАНИЯ**

*Белорусский национальный технический университет,  
г. Минск, Республика Беларусь*

*Научный руководитель: канд. техн. наук, доцент Дробыш А.А.*

Парное программирование – это техника программирования, при которой исходный код создаётся парами людей, программирующих одну задачу, сидя за одним рабочим местом.

В таких случаях два человека сидят плечом к плечу за одним компьютером. Один из них «водитель», его работа заключается в написании кода. Второй делает постоянный ревью кода первого,

чтобы определить тактические и стратегические недостатки в коде, в том числе ошибки в синтаксисе, логике программы, опечатки и реализации, которые не подходят под существующий дизайн системы. После определенного времени программисты меняются ролями, либо меняют пары.

Цель данного подхода – это произвести более качественный и поддерживаемый код. Эта цель вполне может быть реализована, если участники процесса, работающие с кодом, действуют в соответствии со своими ролями.

Также отмечается, что парное программирование часто применяется там, где практикуется Agile, и многим программистам этот подход очень нравится, так как всегда можно рассчитывать на дополнительную помощь и оценку, вместо того чтобы сражаться с проблемами в одиночку.

Самый популярный вопрос, который возникает насчет этого подхода: не будет ли разработка идти медленней, когда два программиста занимаются одной задачей?

Исследования показывают, что работа в паре делает либо с такой же скоростью, как если бы программист работал в одиночку, либо немного, на 15 % медленнее. Но в таких случаях код получается намного качественнее и содержит меньше ошибок, где-то на 60 % меньше, нежели бы программист работал самостоятельно.

Подход парного программирования имеет следующие достоинства:

- Обмен опытом: часто бывает, что сидя в паре люди могут узнать некоторые интересные утилиты для ускорения работы, горячие клавиши, которые раньше не были известны. В любом случае, работа в паре – это всегда учение.

- Знания о системе: так как идет постоянная смена пар, дается возможность изучить разные части системы внутри команды.

- Наставничество: все когда-то с чего-то начинали, программирование не исключение. Как показывает практика, самое простое влияние в проект происходит в процессе парного программирования.

- Больше общения: большую роль играет общение внутри команды, оно помогает выстраивать доверительные отношения.

- Стандарты кодирования: так как участники процесса постоянно передают клавиатуру из рук в руки, идет распространение знаний о стандартах кода в проекте.

- Улучшение дисциплины: сидя в паре, хочется показать свою заинтересованность и уровень подготовки партнеру. В таких ситуациях всегда довольно трудно временно переключиться на соц. сети, чтобы проверить новые сообщения, время двух человек всегда ценнее, чем одного, их работа становится в 2 раза дороже и каждый участник понимает свою ответственность перед другим.

- Сопряжение потока: здесь все довольно просто – оба программиста погружаются в одну задачу и продуктивность возрастает в разы.

Но бывают такие ситуации, когда человек априори не любит работать в паре или команде. Большая часть попыток парного программирования губится одним из перечисленных ниже антипаттернов:

- Наблюдай за Мастером: бывают случаи, когда в паре есть программист, который является гуру своего дела. И когда его менее опытный напарник задает ему какой-либо вопрос, он чаще всего игнорируется либо идет популярный ответ: «Поуггли». Мастер не спешит передавать работу напарнику, а если до этого дело и доходит, то Мастер теряет всякий интерес к работе, так как наблюдать за работой новичка Мастеру не очень интересно.

- Диктатор: один из разработчиков может занимать в паре жесткую позицию по поводу всех решений, которые касаются текущих задач. В такой ситуации не может идти речи о взаимной помощи или обучении в паре. Поэтому важно помнить, что при парном программировании оба разработчика работают на равных, у каждого из них есть свое мнение, и эти мнения им нужно согласовывать.

- Сходи за кофе: пара садится за компьютер. Один из разработчиков берет клавиатуру и начинает писать код. Говорит напарнику: «Пока я пишу код, ты сходи и налей нам кофе». Это нарушает базовую идею о взаимной вовлеченности программистов в процесс.

- Молчаливые партнеры: наверное, самое губительное, что может быть при командной работе – напарники не общаются друг с другом и не комментируют свои действия и решения по ходу работы. При отсутствии обратной связи смысл пары теряется.

- Разделение задач за одним столом: программисты садятся в пару, берут два компьютера за одним столом и начинают параллельно работать. Весь смысл подхода сразу теряется.

- Неудобно сидеть: самая частая причина усталости при работе в паре – неудобное положение клавиатуры и монитора для того, кто сейчас «водитель».

- Свои настройки окружения: каждый раз, когда управление переходит от одного партнера к другому, начинается перенастройка окружения: закладок, шрифта и т.д.

Также многие руководители могут посчитать это невыгодным делом – сажать на одно рабочее место двух работников. Плюс это требует временных затрат, так как каждому нужно согласовать удобный график работы

Таким образом, парное программирование сейчас практикуется и дает неплохие результаты, но подходит оно не для всех программистов, поэтому прежде чем пробовать себя в таком подходе, нужно основательно изучить цели и методику этого подхода и понять, подходит она для дальнейшей работы или нет.

УДК 621.762.4

Шнитко А.В., Мелихов В.А.

## **ТИПИЧНЫЕ ОШИБКИ ИНТЕРФЕЙСА**

*Белорусский национальный технический университет,  
г. Минск, Республика Беларусь*

*Научный руководитель: канд. техн. наук, доцент Дробыш А.А.*

При разработке новых программных продуктов, начинающие специалисты часто делают ошибки при создании пользовательского интерфейса – интерфейса, обеспечивающего передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы. В итоге люди не могут или не хотят пользоваться их сервисом или приложением.

Ниже представлены самые часто допускаемые ошибки интерфейса:

- Концентрация на «впечатляющем» дизайне в ущерб хорошей архитектуре

Здесь ситуация состоит в следующем: разработчик хочет, чтобы его продукт выделялся на фоне остальных. Но здесь разнятся требо-