

Актуальні питання навчання

УДК 004.9

Е.Г. Кириленко

Національний аерокосмічний університет ім. Н.Е. Жуковського «ХАІ», Харків

ОБОСНОВАНИЕ СОДЕРЖАНИЯ ОБУЧЕНИЯ В РАМКАХ МЕТОДОЛОГИИ ПРЕПОДАВАНИЯ ПРОФЕССИОНАЛЬНО-ОРИЕНТИРОВАННОЙ ДИСЦИПЛИНЫ «ЧЕЛОВЕКО-МАШИННОЕ ВЗАИМОДЕЙСТВИЕ»

В статье исследованы особенности разработки пользовательского интерфейса. Приведены подходы к разработке пользовательского интерфейса. Выявлены стандарты разработки ПИ, ориентированные на пользователя. Охарактеризованы этапы разработки пользовательского интерфейса. Приведены цели задачи этапов разработки, а также входные и выходные документы. Определены методы оценки качества ПИ. Выявлены совокупность знаний, умений, навыков, которые могут стать основой для разработки содержания соответствующих тем учебной дисциплины «Человеко-машинное взаимодействие».

Ключевые слова: *программная инженерия, разработка человеко-машинного взаимодействия, подходы к разработке интерфейса пользователя, этапы разработки интерфейса пользователя, содержание дисциплины «Человеко-машинное взаимодействие».*

Введение

Индустрия программного обеспечения – бурно развивающаяся область современной экономики, а программное обеспечение (ПО) – важнейшая составляющая информационных систем, в которых сконцентрирован их интеллект. Основными проблемами, с которыми в настоящее время столкнулась отрасль программной инженерии стали нехватка высококвалифицированных кадров и несоответствие уровня подготовки выпускников требованиям работодателей, дефицит квалифицированных преподавательских кадров. Одним из способов устранения перечисленных проблем является подготовка специалистов по направлению «Программная инженерия».

Необходимость в специалистах по направлению "Программная инженерия" диктуется потребностями информационно-коммуникационной отрасли: открытием ИТ-технопарков, стремительным развитием рынка офшорного и заказного программирования, информатизацией государственных структур, потребностями частного бизнеса.

При проектировании академических учебных курсов по программной инженерии преподаватели используют признанные международные стандарты, такие как профессиональный – SWEBOOK [8], отражающий общепринятое представление о том, что должен знать работник, имеющий степень бакалавра и четырехлетний опыт работы, а также образовательный – SE2004 [2], содержащий руководящие принципы создания учебных планов по программной инженерии для студентов-выпускников. Оба документа особо подчеркивают «инженерную» сторону программной инженерии, направленной как на

содержание типичного курса, так и на понимание студентами сущности изучаемой дисциплины.

В соответствии с рекомендациями, приведенными в SE2004 в программу подготовки бакалавров на младших курсах целесообразно включать учебные дисциплины, ориентированные на программную инженерию. Такой подход к организации подготовки позволит студентам с начала обучения мыслить как разработчик программного обеспечения, концентрировать внимание на решаемой проблеме, изучать требования и особенности проектирования перед созданием кода, думать о процессах, работать по итеративной схеме и использовать преимущество других методов программной инженерии, а также чувствовать, что они обучаются по выбранной специальности [2].

Одной из базовых учебных дисциплин по программной инженерии является «Человеко-машинное взаимодействие». Дисциплина покрывает различные темы, связанные с проектированием и оценением пользовательского интерфейса, а также некоторые основы психологии.

Основной проблемой разработки учебной дисциплины является ее междисциплинарный характер содержания, требующий интеграции профессиональных знаний из разных предметных областей.

1. Анализ исследований и публикаций

Одним из средств реализации содержания образования является учебная дисциплина, которую можно определить как педагогически адаптированное содержание какой-либо области деятельности.

Анализ научной и учебно-методической литературы по вопросам разработки программного обеспечения показал, что менее всего освещены вопросы, связанные с человеко-машинным взаимодействием.

В рекомендациях по преподаванию программной инженерии и информатики в университете к базовым курсам относится курс SE212 «Человеко-машинное взаимодействие». Он является частью вводных блоков по программной инженерии. Студенты должны прослушать этот курс, имея базовые знания по введению в программную инженерию.

Область знания человеко-машинное взаимодействие покрывает темы:

- технология разработки (CMP.ct),
- конфигурация времени выполнения и интернационализация (CMP.ct.8),
- среды разработки графического интерфейса пользователя (CMP.ct.2) модуля CMP;
- инженерные основы программного обеспечения (FND.ef) модуля FND;
- групповая динамика и психология (PRF.psy) модуля PRF;
- основы моделирования (MAA.md), типы моделей (MAA.tm), анализ качества модели (MAA.gfd) модуля MAA;
- проектирование человеко-машинного интерфейса (DES.hci) модуля DES;
- терминология и основы верификации и аттестации программного обеспечения (VAV.fnd), метрики и измерения (VAV.fnd.4), рецензия кода (VAV.rev), инспекция (VAV.rev.3), тестирование атрибутов качества (VAV.tst9), тестирование и оценка человеко-машинного интерфейса (VAV.hct) модуля VAV;
- обеспечение качества продукта (QUA.pda), обеспечение атрибутов качества продуктов (QUA.pda.6) модуля [QUA] [2].

2. Постановка задачи

Согласно рекомендациям по преподаванию программной инженерии, бакалавр должен обладать следующими навыками:

- оценивать пользовательские интерфейсы, используя эвристическое оценивание и методы наблюдения за пользователем;
- проводить простые формальные эксперименты по оценке эргономических гипотез;
- применять ориентированное на пользователя проектирование и принципы эргономики при проектировании широкого круга программных пользовательских интерфейсов.

Как видно, к перечисленным навыкам относятся навыки проектирования и оценивания пользовательского интерфейса, а также эргономические аспекты его разработки.

Таким образом, актуальной является задача теоретического обоснования целей и содержания обучения по дисциплине «Человеко-машинное взаимодействие», которая входит в программу подготовки бакалавров по специальности «Программная инженерия».

3. Разработка пользовательского интерфейса

3.1 Анализ подходов к разработке пользовательского интерфейса

В системе человек-компьютер между человеком и компьютером находятся средства отображения информации, поэтому возникает проблема проектирования средств отображения информации и средств взаимодействия человека с компьютером, т.е. проблема проектирования пользовательского интерфейса.

В условиях постоянного роста количества разрабатываемого программного обеспечения и смены его версий разработка удобного в использовании ПИ позволит:

- повысить конкурентоспособность ПО,
- снизить стоимость разработки ПО,
- увеличить аудиторию использования ПО,
- увеличить удовлетворенность пользователей,
- уменьшить затраты на обучение и поддержку пользователей.

Исследования в области человеко-машинного взаимодействия показывают, что любой пользовательский интерфейс должен обеспечивать выполнение следующих четырех функций:

- управление компьютером путем действий пользователя: инициация, прерывание, отмена компьютерных процессов и др.;
- ввод данных, осуществляемых пользователем, и отклик системы;
- отображение данных, вводимых пользователем;
- поддержка пользователя в процессе деятельности, которая включает в себя обратную связь и сбор информации об ошибочных или случайных действиях пользователя [5].

Хорошо спроектированный пользовательский интерфейс должен соответствовать представленным ниже принципам:

- иметь низкий порог вхождения, то есть способствовать быстрому освоению пользовательского интерфейса, формированию у пользователя устойчивых навыков;
- обеспечивать ввод информации естественным образом, не демонстрируя пользователю ход вычислительного процесса;
- удовлетворять рабочие потребности пользователя, не заостряя его внимание на процессе обработки данных [4].

Существуют различные подходы к разработке пользовательского интерфейса.

Подход, ориентированный на пользователя (User-Centered Design) [5] характеризуется следующими признаками:

- активным вовлечением пользователей в процесс проектирования и тестирования программного продукта;
- четким пониманием пользовательских требований и задач;
- оптимальным распределением функций между пользователями и технологиями.

Эксперты в области проектирования взаимодействия [6] выделили такие основные принципы, характеризующие успешный процесс разработки ПИ:

- фокусировка на пользователе,
- активное участие пользователей,
- эволюционное развитие системы,
- простое решение дизайна,
- прототипирование,
- оценка использования ПИ в контексте,
- явные и осознанные этапы разработки,
- профессионализм,
- целостный дизайн,
- настройка процессов,
- обоснованная ориентация на пользователя.

Применение данного подхода в процессе разработки ПИ способно привести к достижению высоких показателей в области удобства использования, к сокращению расходов на разработку и повышению эффективности продукта как в отношении бизнеса (дополнительная прибыль), так и в удовлетворенности пользователей (повышение лояльности к продукту и разработчику).

Подход, ориентированный на деятельность пользователя (Activity-Centered Design) [5], позволяет фокусироваться не на конечных целях и предпочтениях пользователя, а на исследовании, каким образом пользователи ведут себя при выполнении конкретных задач. При этом, деятельность определяется как действия и решения, выполненные с определенной целью. Выполнение действий заканчивается, когда лицо, совершающее действие, либо внешнее воздействие останавливает его.

В данном подходе дизайнеры (эксперты) используют исследования для получения информации о способностях пользователей разбираться по существу вопроса. Методики (методы) наблюдения и интервью представляют собой типичные подходы, необходимые для того, чтобы узнать как можно больше о поведении пользователей. Нанеся на карту деятельность (действия) пользователей и задачи, дизайнер (эксперт) может определить недостающие задачи для деятельности, чтобы решение общей задачи было легко выполнимым, и таким образом,

проектные решения, основанные на выполнении этих задач, могли стать более простыми и понятными для пользователей.

Этот подход базируется на общепсихологической теории деятельности, разработанной советским психологом А.Н. Леонтьевым, рассматривающим человека в контексте взаимодействия с окружающей средой. Недостатком этого подхода, по мнению А. Купера, является то, что это подход не предполагает учет целей (мотивов) и потребностей пользователей ПИ [1].

Подход, ориентированный на задачи пользователей (Task-Centered Design). Согласно этому подходу к разработке ПИ тот интерфейс хорош, в котором эффективно выполняются задачи пользователей.

Любую задачу можно рассматривать как цепочку действий на пути её достижения, причем для решения могут использоваться разные методы и способы. Дело дизайнера интерфейсов, согласно идеям подхода – выбрать наиболее эффективное решение задачи и обеспечить её выполнение.

В отличие от потенциальных особенностей пользователей, число задач конечно и более предсказуемо при планировании. Это позволяет сузить временные рамки в процессе работы над проектом, а также делает менеджмент более удобным и эффективным.

Недостатком подхода является то, что он не позволяет определить, какое именно количество решаемых продуктом задач является необходимым и достаточным. Всегда есть задачи, которые система (ещё) не решает, а согласно подходу все задачи нужно встроить в ПИ, поскольку, чем больше задач будет решать ПО, тем оно будет лучше. Таким образом, появляется бесконечный рост функциональности (т.н. creeping featurism) [5].

Целеориентированный подход (Goal Centered Design) [5]. В основе данного подхода лежат конечные цели пользователей, которые должны быть ими достигнуты посредством взаимодействия с программным продуктом. Согласно этому подходу, пользователи взаимодействуют с ПИ для удовлетворения личных потребностей, иначе – целей. Выявив эти потребности и сравнив их с задачами пользователей, дизайнер ПИ получает возможность сделать работу пользователей ПО более эффективной.

Данный подход включает шесть стадий разработки ПИ: исследование, моделирование, разработка требований, определение общей инфраструктуры, детализация и сопровождение. Эти стадии соответствуют пяти видам деятельности, составляющим процесс проектирования взаимодействия: понимание, абстрагирование, структурирование, отображение, детализация, но с более выраженным акцентом на моделирование поведения пользователей и определении поведения систем. Таким образом, полученные решения соответствуют потребностям и це-

лям пользователей с одной стороны, а также бизнес требованиям и технологическим ограничениям – с другой стороны.

Подход, ориентированный на данные (Data Centered Design) [5]. Проектирование интерфейса поддерживает такую модель взаимодействия пользователя с системой, при которой первичными являются обрабатываемые данные, а не требуемые для этого программные средства. Основное внимание пользователя в этом случае концентрируется на тех данных, с которыми он работает, а не на поиске и загрузке необходимого приложения. При использовании этого подхода основным программным объектом является документ, который представляет собой некоторое абстрактное устройство хранения данных, используемых для выполнения заданий пользователей и их взаимодействия. Документ должен быть доступен как различным приложениям, используемым для его обработки, так и всем взаимодействующим пользователям.

Подход, ориентированный на опыт взаимодействия (User Experience Design) [5]. Термин опыт взаимодействия широко применяется в информационных технологиях для описания субъективного отношения, возникающего у пользователя в процессе использования, как информационной системы в целом так отдельной ее части (веб-сайта, приложения и пр.).

Разработчики уделяют значительное внимание изучению и проектированию опыта взаимодействия на всех этапах создания продукта, начиная с самого раннего – этапа планирования.

Еще более важную роль опыт взаимодействия играет в интернете. Во многом это обусловлено спецификой глобальной сети и тем, каким образом пользователи с ней взаимодействуют. С точки зрения пользователя, минимальной структурной единицей сети интернет является веб-страница, которая часто является инструментом самообслуживания. В данном контексте, это означает, что пользователь лишен какой-либо посторонней помощи (инструкций, учебных семинаров, возможности обратиться в службу поддержки и пр.).

Итеративный подход (Agile) [5]. В итеративном подходе к разработке пользовательского интерфейса используется быстрое проектирование, предусматривающее создания быстрых прототипов ПО и утверждение их у обширной группы представителей заказчика, заинтересованных сторон, разработчиков и дизайнеров.

Процесс быстрого создания прототипов подразумевает множественные итерации в рамках рабочего процесса, состоящего из трех этапов: создание прототипа, основываясь на стандартах в области проектирования и принципах проектирования ПИ; оценку прототипа; улучшение прототипа.

Данный подход позволяет получить быструю обратную связь, что значительно улучшает качество разработки конечного продукта и уменьшает количество изменений в процессе разработки.

3.2 Этапы разработки ПИ

Процесс разработки ПИ основывается на стандартах:

Ergonomics of human-system interaction Part 210: Human-centred design for interactive systems (ISO 9241-210:2010). Эргономика взаимодействия «человек-система». Часть 210. Проектирование, ориентированное на человека, для интерактивных систем.

ISO/IEC TR 9126-3 (2003): Software engineering – Product quality — Part 3: Internal metrics. Программирование. Качество продукта. Часть 3. Внутренние метрики.

ISO/IEC 9126-4 (2004): Software engineering – Product quality – Part 4: Quality in use metrics. Программирование. Качество продукта. Часть 4. Качество при использовании метрик.

Стандарт ISO 9241-210:2010 «Процесс разработки интерактивных систем, ориентированных на пользователей» содержит указания и рекомендации по организации процесса проектирования интерфейсов, ориентированных на пользователя (UCD – User Centered Design) и органичному встраиванию этого процесса в общий процесс производства ПП. В стандарте описаны этапы разработки ПИ и методы, необходимые для определения контекста использования продукта и выявления требований пользователей и заказчиков к системе; проектирования, дизайна, прототипирования и оценки удобства использования ПИ. Рассмотрим этапы разработки ПИ.

Анализ контекста использования. Цель данного этапа обеспечить эффективное использование методов в процессе дизайна и разработки ПИ и обеспечить постоянное влияние этих методов на самых ранних стадиях проекта.

Прежде чем приступать к разработке и построению любой системы, необходимо выяснить, какие проблемы потребители или пользователи хотят решить, и как они привыкли работать.

Существует ряд ключевых вопросов, которые следует поставить на этапе анализа информации от пользователей. Первый этап – действия по сбору и анализу информации – может быть разбит на пять шагов: определение профиля пользователей; анализ стоящих перед ними задач; сбор требований, предъявляемых пользователями; анализ рабочей среды пользователей; соответствие требований пользователей стоящим перед ними задачам.

На этапе анализа используют методы: интервью с заинтересованными в проекте лицами; контекстные исследования; этнография (наблюдения и эксперимент), работа с представителями пользова-

телей. Выходным документом является спецификация, в которой описано Общее описание проектируемой системы, необходимое для общего понимания продукта проектной группой: видение продукта, бизнес цели заказчика, описание пользователей (роли или персонажи), сценарии использования (use cases). Для контроля качества используют аудит (коллегами, заказчиком, заинтересованными лицами).

Проектирование интерфейса. Исходными данными для проектирования ПИ являются спецификации, разработанные на предыдущем этапе. На этом этапе решаются следующие задачи: определяется тип приложения и способы управления ПИ; разрабатывается физическая структура окон, страниц и элементов управления и определяется система навигации между ними; определяются объекты, данные, действия из контекстных сценариев и задач, стоящими перед пользователями, а также списки тематических категорий и списки инструментов; разрабатывается схематичное изображение содержания экранов, форм.

Выходными документами являются диаграммы навигации, действий, структурные схемы. Для контроля качества используют стандарты проектирования, шаблоны интерфейса, аудит.

Дизайн интерфейса. Исходными данными для дизайна являются спецификации, диаграммы, структурные схемы. На этом этапе решаются задачи: разрабатываются метафоры и идиомы объектов и их визуальное представление или используются стандартные из соответствующих руководств по стиливому оформлению; осуществляется визуальный дизайн ПИ (цвет, шрифт, графика и др.); осуществляется компоновка элементов страницы или формы; определяется представление контента страниц, экранов, форм и инструментов (калькулятор, календарь и др.); разрабатывается макет страницы, максимально приближенный к внешнему виду готового ПИ.

Разрабатываемый интерфейс должен соответствовать как стандартам поставщика интерфейса, так и внутрикорпоративным стандартам, которые связаны с интерфейсом. Выходным документом является макет страницы и инструкция по стилю. Для контроля качества используют экспертный аудит.

Разработка прототипа. Исходными данными для разработки прототипа являются спецификации, диаграммы, структурные схемы, макеты. На этом этапе разрабатывается бумажный или интерактивный прототип.

Основными преимуществами бумажного прототипа являются: потенциальные проблемы с ПИ можно обнаружить уже на ранних этапах дизайна еще до того, как был написан код; устанавливается общение между дизайнерами и пользователями; бумажные прототипы легко и быстро создаются и изменяются, что позволяет быстро проводить тесты в

несколько этапов; для метода требуются минимальные ресурсы.

Для того, чтобы протестировать интерфейс более детально, используют страницы, на которых уже присутствуют меню, списки, диалоговые окна и прочие элементы интерфейса. Пользователь просто указывает пальцем или карандашом на нужный элемент, и вписывает текст, где требуется, и таким образом эмулирует работу с системой.

Основными целями разработки интерактивного прототипа являются: демонстрация основы ПИ заказчику; оценка качества ПИ.

Для реализации прототипа ПИ имеется большое количество инструментальных средств разработки [10]. При выборе инструментальных средств можно воспользоваться критериями: простота изучения основных функций программы; гибкость программы; наличие репозитория с достаточным количеством полезных трафаретов, шаблонов и элементов; возможность вносить быстрые изменения; возможность совместной работы; стоимость лицензии.

Выходным документом на этом этапе является прототип ПИ. Для контроля качества используется аудит и пользовательское тестирование.

Оценка качества ПИ (usability). Цель данного этапа оценить usability пользовательского интерфейса. Исходными данными для оценки являются спецификации, диаграммы, структурные схемы, макеты, прототип ПИ. Тестирование usability ПИ проводится для того, чтобы выявить проблемы и оценить качество работы продукта, выяснить, насколько он эффективен, продуктивен и довольны ли им пользователи.

Тестирование на удобство применения осуществляется на различных этапах разработки продукта (образец дизайна на бумаге, программные прототипы и конечный продукт), чтобы обеспечить обратную связь с пользователями. Это помогает совершенствовать весь проект в целом, сокращает количество ошибок, проводит сравнительный анализ продуктов и версий, а также подтверждает соответствие продукта предъявляемым ему требованиям.

Методы оценки удобства применения, зависят от целей и задач, стоящих перед продуктом. Как правило, тестирование на удобство применения включает два типа методов оценки:

- количественные методы оценки работы, подразумевающие подсчет действий, определение полноты выполнения задачи, подсчет времени, затраченного на это выполнение, ошибок и обращений за помощью.

Преимущества методов:

- возможность использовать измеряемые критерии, которые можно использовать при последующих итерациях для оценки продвижения к цели;

- возможность использования статистических методов для подтверждения достоверности результата тестирования перед заинтересованными лицами, которые принимают решения на основании объективных данных;

- сокращается вероятность того, что на результат повлияет субъективность конкретного разработчика ПИ;

- повышается вероятность, что полученные результаты отражают объективную картину и могут быть перенесены на пользовательскую аудиторию.

- существует наглядный числовой критерий проверки полученных результатов. Например, сколько пользователей столкнулось с одной и той же проблемой;

- простота и удобство расчетов;
- отсутствие параметров в модели позволяет проводить оценочные сравнение двух разных вариантов интерфейса;

- дает прогноз времени работы пользователя с данным вариантом интерфейса;

- модель не требует создания рабочего прототипа;

- анализ по этой модели может быть автоматизирован.

Недостатками методов являются:

- ориентация на средних пользователей и отсутствие учета особенностей работы новичков и специалистов, а также индивидуальных различий пользователей;

- отсутствие учета возникновения случайных ошибок в работе;

- отсутствие учета того, что в процессе работы происходит научение, а при простое – забывание;

- отсутствие учета насколько представляемая интерфейсом информация сложна для понимания пользователем;

- отсутствие учета насколько интерфейс отвечает требованиям пользователей и их ожиданиям.

Качественные, субъективные методы включают сбор устных и письменных сообщений пользователей об их восприятии, мнениях, суждениях, предпочтениях, а также степени удовлетворенности от системы и их собственной выполненной работы.

К преимуществам качественных методов относят:

- открытость процесса тестирования, что способствует исследованию новых идей и проникновению в сущность проблемы;

- возможность обсуждения проблем ПИ с пользователями;

- возможность провести тестирования меньшим количеством респондентов и не требует знания формальных статистических методов анализа и об-

работки данных, что позволяет уменьшить стоимость тестирования.

Качественный подход фактически сопряжен с меньшим риском, чем количественный, в тех случаях, когда тестирование было проведено некорректно [3].

Как правило, при тестировании удобства использования пользовательского интерфейса используются некоторые эвристические критерии и характеристики, которые заменяют точные оценки в классическом тестировании программных систем. Якоб Нильсен выделил 10 эвристических характеристик удобного пользовательского интерфейса, которые, по его мнению, должны проверяться при тестировании удобства и простоты использования интерфейса: наблюдаемость состояния системы, соотношение с реальным миром, пользовательское управление и свобода действий, целостность и стандарты, помощь пользователям в распознавании, диагностике и устранении, предотвращение ошибок, распознавание, а не вспоминание, гибкость и эффективность использования, эстетичный и минимально необходимый дизайн, помощь и документация [7].

Количество измеряемых показателей в конкретном тесте может быть довольно велико, но все они, как правило, сводятся к набору из 3 базовых характеристик: эффективность, производительность, удовлетворенность ПИ. Выходной информацией этапа является отчет по оценке usability ПИ.

Анализ процесса разработки пользовательского интерфейса позволяет выделить совокупность знаний, умений и навыков, необходимых будущему специалисту для разработки пользовательского интерфейса. Будущие специалисты в области программной инженерии должны *знать*: основы человеко-машинного взаимодействия; обоснование человеко-машинного взаимодействия с точки зрения психологии и когнитивных наук; концептуальные модели и метафоры; методы оценивания пользовательских интерфейсов и методы наблюдения за пользователем (эвристическая оценка, когнитивные просмотры); круг задач, ориентированных на пользователя, которые анализируются для этапов проектирования и прототипирования ПИ; эргономические принципы проектирования ПИ; методы выполнения формальных экспериментов по оценке эргономических гипотез; этапы разработки пользовательских интерфейсов.

Будущий специалист в области программной инженерии должен *уметь*: применять, ориентированное на пользователя проектирование ПИ; применять принципы эргономики в процессе проектирования широкого круга пользовательских интерфейсов; обосновывать проектные решения; проектировать окна, меню и команды; реализовывать интернационализацию и локализацию ПИ; оценивать пользовательские интерфейсы, используя эвристи-

ческое оценивание и методы наблюдения за пользователем; оценивать ПИ, используя видеозапись; реализовывать бумажный и интерактивный прототип; проектировать ПИ методом быстрого прототипирования; проводить простые формальные эксперименты по оценке эргономических гипотез; выбирать архитектуру пользовательских интерфейсов и API; использовать на практике целесообразные шаблоны проектирования пользовательских интерфейсов; разрабатывать концептуальные модели и метафоры; использовать цветовые схемы, шрифты, звук, анимацию для проектирования ПИ; проектировать и реализовывать голосовой ввод/вывод, систему помощи; сообщения об ошибках.

Будущий специалист должен сформировать навыки совместной разработки ПИ.

Выводы

Разработка пользовательского интерфейса представляет собой междисциплинарную деятельность, которая требует усилий многофункциональной группы: специалиста по анализу требования, проектировщика, дизайнера, художника, специалиста в области когнитивной психологии кодировщика, специалиста по тестированию usability ПИ и др.

Обзор процесса разработки ПИ позволил выявить совокупность знаний, умений, навыков, которые могут стать основой для разработки содержания соответствующих тем учебной дисциплины «Человеко-машинное взаимодействие».

Проблема заключается в междисциплинарном характере содержания курса, требующем интеграции знаний в областях программной инженерии, психологии, социологии, антропологии, промышленного дизайна.

Список литературы

1. Купер А. Об интерфейсе. Основы проектирования взаимодействия: пер. с англ. / А. Купер. – СПб.: Символ-Плюс, 210. – 688 с.
2. Рекомендации по преподаванию программной инженерии и информатики в университетах. – М.: ИНТУИТ.РУ «Интернет-Университет Информационных Технологий», 2007. – 462 с.
3. Унгер Р. –дизайн. Практическое руководство по тестированию опыта взаимодействия / Р. Унгер, К. Чендлер. – СПб.: Символ-Плюс, 2011. – 336 с.
4. Bevan N. A proposed standard for consumer product usability / N. Bevan, R. Schoeffel // UAHCI, New Orleans, Augus. – 2001.– №12.– С. 123-126.
5. Human-Computer Interaction / A. Dix, J. Finlay, G. Abowd, R. Beale. – Prentice-Hall, New Jersey, 1993. – С. 51-61.
6. Hyatt L. A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality / L. Hyatt, L. Rosenberg, 2001.
7. Nilsen J. / Usability inspection Methods / J. Nilsen, Robert Mack. – New York: Wiley. 1994.
8. SWEBOOK [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.swebok.org>.
9. [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.webmascon.com/archive/>.
10. [Электронный ресурс]. – Режим доступа к ресурсу: <http://www.adaptivepath.com/ideas/rapid-prototyping-tools-revisited>.

Поступила в редколлегию 2.03.2012

Рецензент: д-р техн. наук, проф. И.В. Шостак, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ОБГРУНТУВАННЯ ЗМІСТУ НАВЧАННЯ В РАМКАХ МЕТОДОЛОГІЇ ВИКЛАДАННЯ ПРОФЕСІЙНО-ОРІЄНТОВАНОЇ ДИСЦИПЛІНИ «ЛЮДИНО-МАШИНА ВЗАЄМОДІЯ»

О.Г. Кириленко

У статті досліджено особливості розробки користувальницького інтерфейсу. Наведено підходи до розробки користувальницького інтерфейсу. Виявлено стандарти розробки ПІ, які орієнтовані на користувача. Охарактеризовано етапи розробки користувальницького інтерфейсу. Наведено цілі та завдання етапів розробки, а також вхідні і вихідні документи. Визначено методи оцінки якості ПІ. Виявлено сукупність знань, умінь, навичок, які можуть стати основою для розробки змісту відповідних тем навчальної дисципліни «Людино-машинна взаємодія».

Ключові слова: програмна інженерія, розробка людино-машинної взаємодії, підходи до розробки інтерфейсу користувача, етапи розробки інтерфейсу користувача, зміст дисципліни «Людино-машинна взаємодія».

GROUND OF TEACHING SYLLABUS WITHIN THE FRAMEWORK OF METHODOLOGY TEACHING OF THE PROFESSIONALLY-ORIENTED COURSE SYLLABUS FOR "HUMAN-COMPUTER INTERACTION"

Ye.G. Kirilenko

The article analyses details of user interface (UI) development, providing the necessary information about development approaches and standards for user-friendly interfaces. The author sums up UI development stages as well as their tasks and aims, supplies readers with input and output documents, and introduces possible UI quality assessment procedures and techniques. Expertise, knowledge and skills revealed by the article can be incorporated to form the course syllabus for "Human-Computer Interaction".

Keywords: software engineering, human computer interaction, user interface development approaches, user interface development stages, course syllabus for "Human-Computer Interaction".