

# Отклики на “клики”

Обсуждение статьи “Испытание “щелчками”

## Буря в “греческом зале”

Опубликованная в прошлом номере нашего журнала небольшая статья независимого аналитика **Joe Greco** “Испытание “щелчками” вызвала на удивление бурную реакцию читателей и оживленные дискуссии на сапровских форумах. Напомним, что речь в статье шла о сравнении количества шагов при построении одних и тех же моделей в четырех различных CAD-системах: *Autodesk Inventor 6*, *Solid Edge 14*, *SolidWorks 2003* и *Pro/E Wildfire*. Принцип подсчета был следующим: каждый “клик” кнопкой мышки (обычный или двойной), любое нажатие клавиши (включая *Enter*), выбор иконки меню или элемента геометрии засчитывались как один шаг.

Чтобы расставить все точки над “ё”, мы решили обратиться к специалистам, для которых доскональное знание указанных пакетов является профессиональным требованием. В качестве экспертов выступили: **Евгений Кухаренко** (зам. директора Академии САПР и ГИС при РПК), **Константин Костромин** (менеджер по работе с бизнес-партнерами *Solid Edge* в России, *EDS*), **Иво Липсте** (вице-президент латвийской компании *COLLA*), а также **Дмитрий Мотовилов** (компания *PTS*) и **Александр Курилов** (ООО “ИНЖЕНЕРНЫЙ КОНСАЛТИНГ”). Всех экспертов наша редакция благодарит за сотрудничество.

Но прежде чем говорить о конкретных результатах, хотелось бы остановиться на некоторых критических замечаниях, общих для многих отзывов. В первую очередь речь идет о корректности предложенного сравнения. В наиболее жесткой форме это замечание, пожалуй, сформулировал Евгений Кухаренко: “Г-н *Greco* предполагает, что статистика по количеству щелчков мышью может служить достаточным основанием для характеристики программной системы. Это интересный формализм, но согласитесь, что производство и работа конструктора не предполагают необходимость подсчета “щелчко-дней” наряду с трудоднями. Предположим, что за день конструктор делает порядка 5000 щелчков. Имеет ли значение тот факт, что при работе с каким-либо пакетом делается на 1000 щелчков больше, если конструктор при этом не обладает хорошими навыками работы в САПР? Будет ли в этом случае количество щелчков отражать эффективность использования той или иной системы? Значительно больше на время проектирования объекта влияет принцип проектирования”.

Вопрос о “правильной” методике сравнения пакетов чем-то сродни вопросу о смысле жизни: на него существует множество ответов, но нет ни одного, с которым были бы согласны все. На наш взгляд, право на существование имеет любая корректная методика. Важно лишь чтобы, во-первых, тестирование проводилось

непредвзято и со всей возможной объективностью, а во-вторых, чтобы и автор и читатели понимали, что именно и каким образом тестируется, – и делали из этого правильные выводы.

В этой связи хотим еще раз обратить внимание читателей на то, как сформулировал задачу своего исследования сам *Joe Greco*: “Некоторое время назад я тестировал *Pro/ENGINEER Wildfire*. Моей целью было проверить утверждения *PTS* о том, что новый пользовательский интерфейс пакета позволяет выполнить построения за меньшее количество шагов (выделено нами. – *Прим. ред.*). Тогда я пришел к выводу, что по отношению к версии 2001 эти утверждения соответствуют действительности. После этого мне захотелось провести аналогичное испытание и для других конкурирующих *MCAD*-систем”. И далее: “Следует иметь в виду, что здесь решались всего лишь две случайным образом выбранные задачи. Если взять другой пример – результаты будут иными”.

Таким образом, речь идет не о сравнении функциональности пакетов, гибкости и пр., а лишь о минимизации количества шагов при построениях, т.е. об одном срезе “эргономичности” интерфейса, и лишь для двух конкретных задач. С учетом этих оговорок предложенная автором методика представляется нам вполне корректной.

Приведенная цитата опровергает еще одно предположение, часто встречающееся в читательской почте, – что тестовые задачи были умышленно оптимизированы под какой-то пакет. Поскольку первоначально обе эти задачи использовались для тестирования *Pro/ENGINEER*, то было бы логично говорить, что они ориентированы именно на эту систему. Однако подобный вывод не подтверждается результатами тестов (особенно в авторском исполнении).

Что же касается объективности, то, безусловно, опыт и знания автора отразились на результатах и привнесли свою долю субъективизма. Этот недостаток мы и постараемся сейчас исправить с помощью наших экспертов.

## 1 Autodesk Inventor 6

Свою версию решения тестовых задач с помощью пакета *Inventor*, а также и свои замечания предлагает зам. директора Академии САПР и ГИС **Евгений Кухаренко**.

### Задача №1

В этом задании необходимо построить простое скругление переменного типа. Г-н *Greco* справился с этой задачей за 22 шага, мы же решили задачу за 17 шагов.

Сокращение количества шагов произошло следующим образом:

✓ При задании радиуса в последней точке мы использовали один шаг вместо двух, так как *Inventor* сделал эту точку активной.

✓ Для задания остальных радиусов мы воспользовались другим алгоритмом. Так как две предпоследние точки видны в окне, то на ввод значений для них нам потребовалось 4 шага, а для задания радиусов конечных точек – 5 шагов, включая скроллинг. Говоря вообще, алгоритм задания радиусов для точек может быть разный, но мы всегда можем сразу задавать радиусы для трех точек. В итоге вместо 13 шагов нам понадобилось 9.

### Задача №2

В этом задании результатом должна была стать тонкостенная оболочка, эскизом которой служил образмеренный прямоугольник с двумя различными радиусами. Г-н *Greco* справился с заданием за 43 шага, мы же выполнили его за 36 шагов. Для уменьшения числа шагов необходимо помнить о следующем:

✓ При образмеривании эскиза не нужно сначала проставлять размерные линии, а затем редактировать размеры – на это уходит слишком много времени. Для того чтобы редактирование размеров происходило автоматически, надо включить опцию *Edit dimension when created* в *Tools* → *Application Options* → *Sketch*. После этого спокойно проставляем размеры: 1-й шаг – выбрали линию; 2-й шаг – показали, где расположим размер; 3-й шаг – указали число; 4-й шаг – подтвердили ввод данных. В итоге при образмеривании эскиза мы затратили 9 шагов вместо 13, при редактировании – 8 вместо 9.

Кстати сказать, пример проектирования конусообразной модели, предложенный г-ном *Greco*, достаточно уникален и редко применяется на практике. Если бы модель была обычной, то на операцию выдавливания затрачивалось бы три (без ввода величины выдавливания – два) шага, а не шесть.

✓ При создании оболочки не стоит поворачивать модель в пространстве: во-первых, она уже повернута, во-вторых, не имеет значения, видна или нет поверхность, которую надо удалять, – её можно выделить и “вслепую”, без каких-либо проблем. В итоге на операцию создания оболочки (*Shell*) мы затратили 5 шагов вместо шести.

Предварительный итог: 37 шагов вместо 43.

✓ При необходимости еще больше сократить количество шагов можно использовать скрытые резервы *Inventor* (их “открывают” преподаватели и опытные специалисты). Так, если при образмеривании прямоугольника применять стандартные функции *Autodimension* и *Parameters* для задания значений размеров, то на образмеривание понадобится 8 шагов (вместо 13 у г-на *Greco*). В итоге потребуется 36 “щелчков”.

### Общие замечания

На мой взгляд, рассмотренные в цитируемой статье задачи не характерны для САПР среднего уровня, к которым относят *Inventor*. Ведь такие САПР позволяют решать как задачи проектирования отдельных узлов и сборок, в частности определения и решения задач на собираемость объекта проектирования, так и моделирова-

ния различных типов механизмов. Такие примеры хороши лишь для *AutoCAD*, в то время как *Autodesk Inventor*, по моему мнению и мнению моих коллег, должен решать более сложные задачи более простыми способами.

Примером такого упрощения в проектировании и моделировании является адаптивное проектирование, позволяющее связывать между собой зависимостями уже спроектированные ранее объекты с вновь разрабатываемыми. Подобное моделирование позволяет намного сократить количество ненужных “щелчков” только за счет отсутствия необходимости постоянно перемещаться из режима сборки или создания модели в предыдущую модель для просмотра необходимых размеров.

## 2 Solid Edge 14

Результаты, полученные в *Solid Edge 14*, проверил и прокомментировал **Константин Костромин**, менеджер компании *EDS* по работе с бизнес-партнерами *Solid Edge* в России.

### Задача №1

Более оптимальная тактика построения изначально отличается от предложенной г-ном *Greco*. Мы разобьем построение на два этапа: сначала расставим точки, а потом займемся скруглением. Итак, выбор команды “Эскиз” (1 шаг); выбор плоскости эскиза (верхняя грань “коробочки”) – 1 шаг; команда “Точный ввод” и установка базы точного ввода (2 шага); команда “Точка” (1); простановка точек на ребре (смещение в поле + позиция на ребре) – 2 шага на каждую точку, всего 6; “Готово” (1). В итоге эскиз из трех точек создан за 12 шагов.

Далее скруглим ребро, как было задано: команда “Скругление” (1 шаг); вызов диалога режимов скругления, выбор метода (переменный радиус), кнопка *OK* – всего 3 шага; выбор ребра (2); указание радиусов в точках (5 точек по 2 шага – всего 10); кнопка “Результат” (1); кнопка “Готово” (1). Итого – само скругление создано за 18 шагов; всего на задачу ушло 30 “щелчков”.

Вероятно, автор не учел следующего: во-первых, имеются разные способы указания точек перемены радиуса; во-вторых, нажимать кнопку “Готово” не обязательно, так как система сама “сообразит”, что пользователь перешел к другой команде. Предполагаю, что можно и еще “подсократиться”, если настроить интерфейс строго под эту задачу.

### Задача №2

В отличие от некоторых других систем, *Solid Edge* допускает как построение тел по готовому эскизу, так и построение эскиза в процессе построения тела. В частности, мы выберем следующий порядок построения: построим выступ (по ходу построив и образмерив его профиль), добавим уклон, скруглим нужные ребра и получим тонкостенное тело.

Итак, начинаем: команда “Выступ” (1 шаг); выбор плоскости профиля (1); построение произвольного прямоугольника – 1 (я не ошибся, пишу прописью –

один); команда “Умный размер” (1); простановка двух размеров – по 3 шага, включая изменение значения (6 шагов); кнопка “Готово” – окончание построения профиля (1).

Далее: указание высоты выступа (1); направление выступа (1); команда “Уклон” (1); выбор базы уклона (1); указание “уклоняемых” граней (режим “все перпендикулярные”) – 2 шага; угол уклона (1); кнопка “Дальше” (1); направление уклона (1); команда “Скругление” (1); выбор ребер, скругляемых радиусом 6 мм (2); указание радиуса (1); выбор ребер, скругляемых радиусом 3 мм (2); указание радиуса (1); выбор всех ребер нижней грани (2); указание радиуса (1); команда “Тонкостенное тело” (1); указание толщины стенки (1); указание открытой грани (2); кнопка “Результат” (1); кнопка “Готово” (1). Всего – **35** шагов.

Здесь г-н *Greco* использовал методику предварительного построения эскиза до выбора команды “Выступ”. В *Solid Edge* это допустимо, но не обязательно. Автор правильно отметил автоматический выбор поля размера (длины, угла и т.п.), но не обратил внимания на способы выбора объектов (уклон всех перпендикулярных граней, выбор всех ребер данной грани и т.п.), что тоже снижает количество кликов.

Замечу, что все без исключения команды, которые я использовал, изучаются в базовом учебном курсе *Solid Edge*, т.е. их знают начинающие пользователи. На мой взгляд, приведенный автором метод исполнения упражнений более соответствует стилю, принятому в *SolidWorks*, чем в *Solid Edge*, а инструментарий этих систем все же различается.

### 3 SolidWorks 2003

Результаты, полученные в *SolidWorks*, комментирует вице-президент компании *COLLA* **Иво Липсте**.

По задаче №1 можно в целом согласиться с методикой построения, принятой *Joe Greco*. А вот задача №2 может быть решена немного более эффективно. Для

этого произведем следующую последовательность действий:

✓ Войдем в *Sketch mode* (1 шаг) и выберем функцию *Rectangle* (1). Прямоугольник в *SolidWorks 2003* может быть нарисован за 1 шаг, а не за два, как это делал г-н *Greco*. Таким образом, общее количество шагов для создания эскиза сократится с четырех шагов до трех.

✓ Далее воспользуемся функцией *Autodimension*. Выбираем эту функцию (1 шаг) и подтверждаем выбор (1), – в результате видим на экране уже образмеренный эскиз. Делаем двойной щелчок на первом размере; вводим значение; подтверждаем (3 шага); повторяем те же операции для второго размера (3). В сумме для образмеривания эскиза требуется 8 шагов, а не 9.

✓ Выбираем функцию *Fillet* (1); вводим значение (1); создаем два скругления (2); перемещаем курсор в поле для ввода радиуса (1); вводим другое значение (1); создаем второй набор скруглений (2). Выход из *Sketch mode* – лишний шаг автора, так как при последующем выполнении команды *Extrude* это произойдет автоматически. В результате для редактирования эскиза требуется на шаг меньше – 8 шагов вместо 9.

Дальнейшие построения не отличаются от варианта, приведенного г-ном *Greco*.

В итоге мы сэкономили 3 шага, а общее количество шагов составило **34** вместо 37.

### 4 Pro/ENGINEER Wildfire

Результаты, полученные в *Pro/E Wildfire* для первой задачи, коротко прокомментировал **Дмитрий Мотовилов** (компания *PTS*), а для второй – **Александр Курилов** (ООО “ИНЖЕНЕРНЫЙ КОНСАЛТИНГ”).

#### Задача №1

Если выбирать значения радиусов не на экране, а из списка (рис. 1), то на задание пяти значений радиусов понадобится не 15 шагов, а 11. В итоге на выполнение всего задания понадобится всего **19** шагов вместо 23.

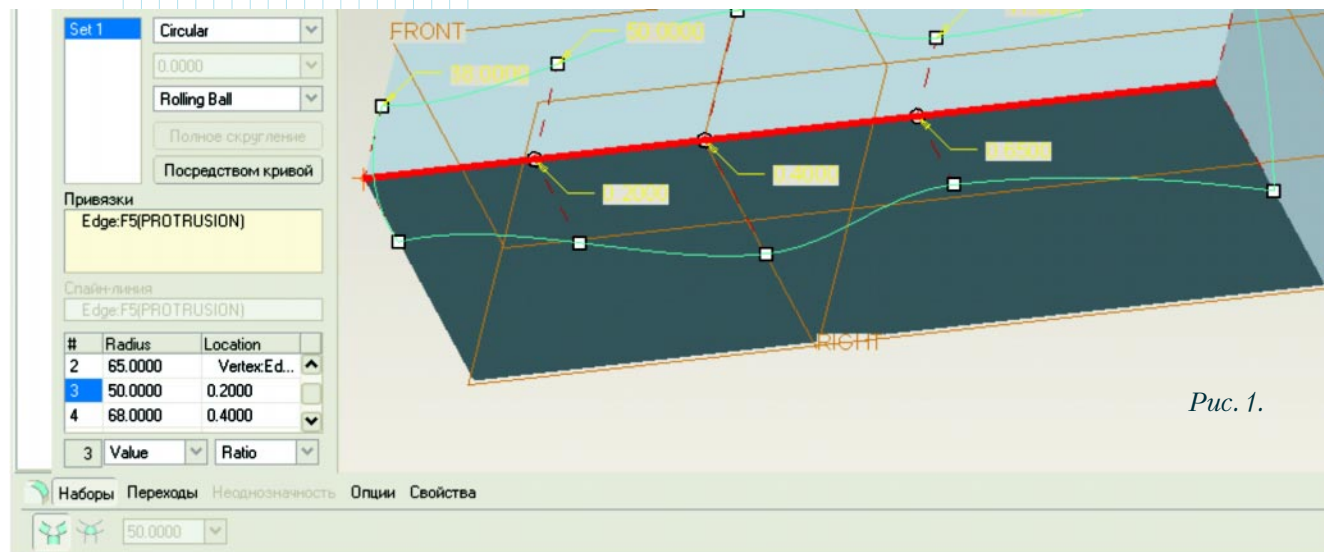


Рис. 1.

Хочется отметить, что по умолчанию в *SolidWorks 2003* ребро делится именно на 5 шагов, и дополнительных действий не требуется. При назначении другого числа точек потребуются дополнительные операции.

### Задача №2

Для решения этой задачи более рациональной является следующая последовательность построений:  
 ✓ Установить режим выбора – *Geometry* (1 шаг); выбрать команду *Extrude* (1); переход в *Sketcher* (1); функция *Rectangle* (1); нарисовать прямоугольник (2); выбрать *Fillet* (1). Для создания скругления надо выбрать две линии (2 шага) плюс еще 6 шагов на остальных углах – итого 8 шагов. Далее выбрать все размеры для редактирования – *Ctrl+Alt+A* (1 шаг); в контекстном меню выбрать *Modify* (1); ввести значения шести размеров (перемещение по списку выполняется автоматически, поэтому получается 6 шагов); завершить редактирование (1); завершить работу с эскизом (1). Масштабирование геометрии в размер экрана производится автоматически, поэтому можно сразу задать высоту с помощью мыши (1 шаг). Остается только завершить операцию (1 шаг). В сумме – 27 шагов.

✓ Выбрать одну боковую поверхность (1 шаг); функция *Draft* (1); выбрать заднюю поверхность, от которой будет строиться уклон (1); с помощью мыши задать необходимый размера уклона (1); завершить операцию (1). В сумме – 5 шагов.

✓ Выбрать одну грань (1); выбрать в контекстном меню *Round Edges* (1); с помощью мыши задать значение радиуса (1); закончить операцию (1). В сумме – 4 шага.

✓ Выбрать поверхность (1); функция *Shell* (1); с помощью мыши задать значение стенки (1); закончить операцию (1). В сумме – 4 шага.

В итоге для построения тестовой детали *Pro/E Wildfire* потребовалось всего **40** шагов вместо 60, заявленных г-ном *Greco*.

Надо подчеркнуть, что в приведенной последовательности нет шагов, связанных с вращением модели в графическом окне. Это вызвано тем, что в *Pro/E Wildfire* реализован эффективный механизм выбора скрытой геометрии. Он ускоряет и облегчает процесс выбора, а главное – устраняет необходимость постоянно вращать модель. А ведь если геометрия модели сложная, процесс её перерисовки может занимать значительное время.

Хочу отметить ещё два момента. Во-первых, не щелчками нужно мерить качество системы, а возможностями построения и редактирования. И в этом случае более легкая и удобная на первый взгляд система может проиграть – из-за того, что та или иная возможность напрямую просто не реализована. Во-вторых, на задачах такого класса, как эти, с упомянутыми *CAD*-системами можно сравнивать не *Pro/E*,

а *Pro/DESKTOP Express* – бесплатный продукт, доступный на сайте компании *PTC*.

### Заключение редакции

Итоговые результаты предыдущего (авторского) и нынешнего (экспертного) тестирования представлены в таблице. Как мы видим, результаты контрольных тестов, проведенных нашими экспертами, не повлияли существенным образом на корректность выводов, сделанных г-ном *Greco* на основе собственного исследования, однако разброс цифр значительно снизился. **В руках специалистов все пакеты показали вполне сопоставимые результаты.** Особенно большие отличия от первоначальных результатов продемонстрировали *Solid Edge 14* и *Pro/E Wildfire*. При выполнении задачи №1 пакетам *Inventor 6* и *Pro/E Wildfire* удалось даже немного потеснить *SolidWorks 2003*, прежде лидировавший в обоих тестах.

**Сводная таблица количества шагов, необходимых для построения тестовых моделей**

|                    | <i>Inventor 6</i> | <i>Solid Edge 14</i> | <i>SolidWorks 2003</i> | <i>Pro/E Wildfire</i> |
|--------------------|-------------------|----------------------|------------------------|-----------------------|
| Задача №1          |                   |                      |                        |                       |
| • <i>Joe Greco</i> | 22                | ~ 45                 | 20                     | 23                    |
| • эксперты         | 17                | 30                   | 20                     | 19                    |
| Задача №2          |                   |                      |                        |                       |
| • <i>Joe Greco</i> | 43                | 44                   | 37                     | 60                    |
| • эксперты         | 37 (36)           | 35                   | 34                     | 40                    |

Один из важных выводов, который можно сделать из сравнения полученных результатов, хорошо сформулировал Евгений Кухаренко. Он пишет о том, что результаты выполнения задач экспертами “... показывают читателям необходимость и преимущества обучения таким сложным программным системам в специализированных учебных центрах – там, где работают истинные профессионалы, отлично реализующие принцип “обучение через изучение”.

А закончить обсуждение хочется ироничным комментарием Константина Костромина: “Если я правильно понял тенденцию сравнений, следом пойдет тест на самый короткий путь курсора мыши. Могу сказать, что наши разработчики уже работают в этом направлении. В частности, интерфейс версии 14 был изменен так, чтобы сократить не только перемещения мыши, но и амплитуду перемещения взгляда пользователя. Результат особенно понравится тем, у кого мышь не оптическая, а механическая. Чистить колесико можно будет гораздо реже”.

Ну что ж, собственно говоря, – почему бы и нет? Соответствующая модель манипулятора типа “мышь” (см. обложку предыдущего номера) редакцией уже запатентована. ☺ 🧠