

## **Разъяснения**

### к Техническим замечаниям к спецификации ECMA-376, предоставленным в Чешский Институт Стандартизации

Поскольку процедуры ISO/IEC и JTC-1 не предусматривают на этапе до 2 сентября 2007 г. возможности подготовки официальных ответов на замечания стороной, внесшей предложение (в данном случае ECMA International, являющаяся A-liaison member JTC-1), и тем более, кем-либо из членов комитета по подготовке стандарта ECMA-376 (Apple, Barclays Capital, BP, The British Library, Essilor, Intel Corporation, Microsoft, NextPage Inc., Statoil ASA, Toshiba, Novell, Библиотека Конгресса США), настоящий документ является неофициальным. Он подготовлен в инициативном порядке Владимиром Габриелем и Владом Шершульским, которым помогли Дмитрий Андреев, Вильфред Громен, Гайдар Магдануров, Дмитрий Мартынов, Грэг Стоун, Алексей Федоров, Кайя Циглик. Документ представляет исключительно личную точку зрения авторов и никак не связан с возможной позицией и оценками компании Microsoft и комитета ECMA-45. Все оценки, суждения и ошибки сделаны исключительно авторами разъяснений исходя из собственного опыта, понимания проблемы и анализа имеющейся документации.

Авторы разъяснений хотели бы подчеркнуть уважение к открытому и подробному обсуждению спецификации ISO/IEC DIS 29500 (ECMA Office Open XML) в Республике Беларусь, выразить удовлетворение тем, что данный стандарт уже нашел применение в высокотехнологичных предприятиях Беларуси, и отметить, что в Республике Беларусь пока не было выявлено каких-либо существенных недостатков данной спецификации. Необходимо также констатировать, что комментируемые «чешские Технические замечания» наряду с несколькими действительно полезными предложениями содержат тенденциозные или основанные на невнимательности суждения.

Дополнительная информация о стандарте, использующих его продуктах и ходе обсуждения может быть получена в следующих источниках:

<http://www.ecma-international.org/publications/standards/Ecma-376.htm> (проект стандарта)

<http://OpenXMLCommunity.org>

[http://www.ecma-international.org/news/TC45\\_current\\_work/OpenXML%20White%20Paper.pdf](http://www.ecma-international.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf)

<http://OpenXMLDeveloper.org>

№ п/п	Раздел	№ параграфа, рисунка, таблицы, примечания	Тип замечания (ge, te, ed)	Суть возражений	Разъяснение к возражению
1.	Office Open XML Overview	-	ge	<p>Не указано, что документ является частью стандарта ECMA в предисловии к части 1 «Fundamentals», соответственно, его статус как нормативного или информативного документа явным образом не определен.</p> <p>Статус документа должен быть уточнен. Если это преимущественно реклама ECMA376, то она не должна включаться в опубликованный стандарт.</p>	<p>Замечание не относится к стандарту. Трудно сказать с определенностью, чем вызвано его появление. Указанный документ нигде не упоминается как часть стандарта и не имеет маркировки ECMA-376 (см. <a href="http://www.ecma-international.org/publications/standards/Ecma-376.htm">http://www.ecma-international.org/publications/standards/Ecma-376.htm</a> )</p> <p>Указанный обзорный документ не является нормативной частью стандарта и приведен в материалах рабочей группы ECMA-45 исключительно в информационных целях для удобства ознакомления с задачами проекта.</p> <p>Указанный обзорный документ кратко описывает структуру предлагаемого проекта и потенциальные области его применения, а также содержит краткое упоминание о рабочей группе по разработке стандарта ECMA-45 (которая, в соответствии с обычной практикой, в случае одобрения проекта будет также заниматься его сопровождением и доработкой). Данная информация по своему содержанию даже отдаленно не является «рекламой», как утверждают авторы замечания, и аналогична информации, присутствующей в некоторых стандартах или общеизвестных комментариях к ним.</p> <p>Следование данному документу не подразумевается, не определяется, не требуется и не влияет на соответствие стандарту.</p>

2.	Office Open XML Overview	§1, p.1	ge	<p>Согласно тексту, предложенный стандарт «был изначально разработан с целью максимального охвата и представления ранее созданного массива текстовых документов, презентаций и электронных таблиц, закодированных в двоичных форматах, определенных корпорацией Microsoft». Такая цель неуместна в процессе стандартизации. Хотя офисное ПО и может поддерживать форматы предыдущих поколений наравне с новыми стандартами, в то же время новые стандарты не должны быть обременены устаревшими особенностями форматов предыдущих поколений.</p>	<p>Замечание прямо не относится к стандарту.</p> <p>Приведенный в кавычках текст является достаточно вольным переводом взятого вне контекста фрагмента из обзорного документа (см. <a href="http://www.ecma-international.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf">http://www.ecma-international.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf</a>) не являющегося частью проекта стандарта, как указано в разъяснении к замечанию 1.</p> <p>В то же время сама по себе задача обеспечения возможности перевода старых документов из бинарных форматов Microsoft Office (равно как и из Word Perfect и других) в современные форматы на основе XML является насущной в целях долговременного хранения и управления жизненным циклом цифровых документов, накопленная база которых в мире составляет около 40 миллиардов единиц хранения. Эта задача прямо поставлена перед поставщиками ПО директивами ряда международных организаций и правительств (см., например, <a href="http://ec.europa.eu/idabc/en/document/6474/5935">http://ec.europa.eu/idabc/en/document/6474/5935</a> ; <a href="http://ec.europa.eu/idabc/en/document/2592/5588">http://ec.europa.eu/idabc/en/document/2592/5588</a>).</p> <p>Не случайно недавние решения государственных органов Дании, Германии, ряда штатов США и др. уже до решения ISO предусматривают возможность использования OpenXML как формата документов государственной практики.</p> <p>Отсутствие приемлемого решения для преобразования документов из бинарных форматов в XML-форматы в рамках формата ODF (ISO/IEC 26300) является одним из подтверждений различных областей применимости спецификаций ISO/IEC DIS 29500 (ECMA Open XML) и ISO/IEC 26300 (OASIS ODF), а также свидетельством целесообразности постановления Open XML под контроль ISO.</p>
----	--------------------------	---------	----	--	--

3.	Часть 1, приложение		te	<p>Ссылка на формат Zip не уточняет версию и дату спецификации, в то время как сама спецификация часто меняется.</p>	<p>В спецификации (Часть 2 “Open Packaging Conventions”) дана явная отсылка на референтное описание: ZIP File Format Specification, Version 6.2.1, PKWARE Inc., 2005. Также указано, что хотя в настоящее время таких планов не имеется, спецификация является открытой, и будущее развитие стандарта может включать возможность использования иных схем архивирования, если ISO/IEC сочтет это целесообразным.</p> <p>Проект стандарта 29500 (см. в частности раздел 9.2 части 2 “Open Packaging Conventions”) указывает, что для целей данной спецификации используется подмножество возможностей архивов Zip. Это подмножество не связано с новыми, расширенными или специальными (например, восходящими к соглашениям Java) особенностями архивации и заведомо будет поддержано в новых версиях спецификаций Zip в целях обратной совместимости.</p> <p>Все дополнительные возможности, которые связаны с упаковкой (например, вопросы поддержки шифрования и цифровой подписи на уровне пакета) или потенциально могут быть связаны с интеллектуальной собственностью третьих сторон, исключены из области отсылки к форматам Zip. Они могут быть реализованы на основе данного проекта стандарта или стандартов ISO и, таким образом, очищены как от особенностей будущих изменений, так и от возможных претензий.</p>
----	---------------------	--	----	--	--

4.	Часть 1, секция 8.6.2	§8.6.2	te	<p>Формат VML объявляется устаревшим, а его поддержка – включенной в предлагаемый стандарт исключительно в целях обратной совместимости. В новых приложениях настойчиво рекомендуется использовать формат DrawingML. Эти положения противоречат части 3 §2.3.1, где утверждается, что вся фоновая информация в WordprocessingML сохраняется с использованием синтаксиса VML.</p>	<p>В замечании дается вольная (расширительная) трактовка положений проекта стандарта, в частности секции 8.6.2 Части 1. DraingML является, безусловно, более «богатым» и более «прогрессивным» форматом векторной графики и рекомендуется проектом стандарта к применению для описания векторных графических изображений. Однако для нескольких простых задач, не связанных непосредственно с созданием сложных изобразительных объектов, средства VML обладают оптимальной выразительностью и могут быть легко реализованы. Это также значительно облегчает достижение обратной совместимости.</p> <p>Авторы разъяснения не располагают информацией, платинируется ли когда либо в будущем исключить VML из стандарта 29500. Однако даже если это произойдет (в неопределенном будущем) несколько простых выразительных тегов могут быть сохранены в другом пространстве имен. Авторы разъяснения признают, что использование DrawingML в данном контексте было бы более элегантным, но и более ресурсоемким решением.</p> <p>Отметим, что для разработчика приложения в принципе сохраняется возможность реализовать только ту часть функциональности VML, которая ему нужна. Такая практика не поощряется в OpenXML, но вполне приемлема в ODF (где именно так реализуется, скажем SVG – с добавлениями и изъятиями).</p> <p>На взгляд авторов разъяснений, национальный орган по стандартизации может, если посчитает нужным, переqualificировать это замечание в ge (general) и внести его в отредактированной форме как рекомендацию для новеллы в следующей версии стандарта.</p>
----	-----------------------	--------	----	--	--

5.	Часть 1, секция 10.1.2	строка 20	te	Текст ссылается на материал части 5, раздел 12. Хотя данный раздел существует, в нем не содержится материалов, на которые ссылается секция 10.1.2. К тому же этот раздел не является нормативным.	<p>Строка 20 Части 1 секция 10.1.2 гласит: «<i>Clause 12 of Part 5 specifies the ability for a markup language to define additional constructs for extensibility</i>». Раздел 12 «<i>Preprocessing Model for Markup Consumption</i>» Части 5 как раз и рассматривает вопрос об интерпретации расширений средствами препроцессора и возможности валидации полученного XML документа. В частности, она разъясняет, как может быть организовано включение новых пространств имен (xmlns) в рамках (и без изменений) стандарта.</p> <p>По мнению разработчиков проекта отсылка является полезной. Однако, поскольку, как справедливо указано авторами замечания, раздел 10.1.2 является информативным а не нормативным, наличие отсылки к нему (также не имеющей нормативной силы) не влияет на нормативное значение стандарта. Если национальный орган стандартизации считает, что эта ссылка является излишней, то замечание можно переквалифицировать в редакционное (ed – editorial) и направить в ISO.</p>
----	------------------------	-----------	----	---	---

6.	Часть 1, секция 12.3.5	-	te	<p>Заявлено, что для хранения произвольных данных, определяемых пользователем, используется двоичный формат. При этом не уточняется, каким пользовательским действием возможно переключить режим хранения этих «определенных пользователем» данных. Без дополнительного определения обеспечение переносимости этих данных невозможно.</p>	<p>Не совсем точный перевод замечания. В оригинале речь идет о возможности использования «custom property part» для хранения двоичных данных, ассоциированных с листом/файлом электронной таблицы (Spreadsheet), если такое решение принято пользователем. В замечании также смешаны понятия об интероперабельном декларировании включения «внешних» или «бинарных» данных (которое целесообразно предусмотреть в стандарте) и интероперабельностью самих «бинарных» данных (которая, конечно, не может требоваться стандартом).</p> <p>По тексту секции 12.3.5 Части 1 речь идет о возможности включения бинарных данных в документ средствами некоторого приложения, поддерживающего Open XML, но, возможно, не только его. Данная ситуация может возникнуть в сценариях, предусматривающих интеграцию традиционных офисных приложений (скажем, OpenOffice.org или Microsoft Office) с некоторыми бизнес-приложениями (line of business applications). Однако вполне возможны сценарии включения в документы и иных бинарных данных самой произвольной природы (например, показаний медицинских приборов и пр.).</p> <p>Проект стандарта не делает никаких спекулятивных предположений о природе и назначении таких данных, а также о поведении, которого хотел бы добиться от приложения пользователь или разработчик. Описывается только интероперабельный механизм индикации включения данных в документ. Соответственно, стандарта не должен содержать деталей поведения, выходящего за его рамки.</p> <p>Хотя решение о поддержке данной опции принято экспертами ECMA-45 исключительно исходя из внутренней логики проекта стандарта 29500, его можно, в определенной мере рассматривать, как реакцию на недостаточность подобных средств в стандарте 26300.</p>
----	------------------------	---	----	---	--

7.	Часть 1, секция 15.2.12	-	te	<p>Не приведены ссылки на конкретные версии спецификаций TrueType или OpenType. Также, поскольку спецификации TrueType и OpenType отличаются, стоило бы ссылаться на них различными способами, а не по общему имени «application/x-font-ttf».</p>	<p>Как известно, спецификация OpenType расширяет спецификацию TrueType. В проекте стандарта 29500 OpenXML не содержится никаких положений, требующих отсылки или особого поведения в отношении конкретной версии спецификации шрифта.</p> <p>Естественно полагать что большинство реальных офисных приложений сможет работать с разными версиями спецификаций шрифтов (и это составляет важную часть реальной, а не декларативной интероперабельности).</p> <p>В то же время, если появятся новые версии описаний шрифтов без обратной совместимости, или если разработчик конкретного приложения предпочтет ограничиться конкретными (новыми или старыми) версиями спецификаций, должна сработать схема подстановки подходящих шрифтов (где возможно, на основе Panose System, см. Часть 4 секция 2.8.2.13).</p> <p>См. разъяснение к замечанию 73.</p>
----	-------------------------	---	----	---	--



8.	Часть 1, секция 15.2.14	-	te	<p>Представляется неудовлетворительным решение, согласно которому настройки принтеров хранятся в зависящих от операционной системы двоичных форматах типа структур DEVMODE. Это вызывает значительную озабоченность с точки зрения безопасности (двоичные структуры DEVMODE загружаются непосредственно в область памяти драйвера устройства), а также усложняет кросс-платформенную реализацию. Исключается переносимость настроек принтера.</p> <p><i>Известны альтернативные решения для описания настроек принтера средствами XML без использования двоичного представления. В частности, спецификация XPS, разработанная корпорацией Microsoft, определяет разметку PrintTicket следующим образом: «PrintTicket – подмножество XML, предоставляющее настройки печати согласованным, доступным и расширяемым образом». Аналогичные свойства для настроек печати в OOXML были бы предпочтительнее контейнеров с двоичными форматами.</i></p>	<p>Данная возможность является зависимой не от операционной системы (в проекте приведены примеры для разных операционных систем), а от аппаратного обеспечения принтера. Как правильно указывают сами авторы замечания, существует (достаточно давно предложенный Microsoft) открытый, универсальный, расширяемый, основанный на XML язык описания настроек принтеров. Проект стандарта OpenXML поддерживает эту тенденцию, предлагая для настроек принтеров XML-средства, но разработчики принтеров могут посчитать нужным использовать дополнительные принтер-специфичные настройки. Тогда они должны сохраняться согласно Части 1, секции 15.2.14 (Printer Settings Part). Разработчики проекта стандарта 29500 разделяют озабоченность уязвимостями аппаратно-программного обеспечения принтеров и полагают, что со временем описание настроек принтеров посредством XML расширится и станет общепринятым, и подтверждают намерение отразить это в следующих версиях стандарта.</p> <p>Альтернативное решение (запрет использования двоичных структур в описании настроек принтеров) может привести к неоптимальной работе или ошибкам работы некоторых принтеров под управлением отдельных приложений (это особенно вероятно для высокопроизводительного профессионального печатного оборудования) и, заведомо, только снизит переносимость и совместимость решений.</p>
----	-------------------------	---	----	---	---

9.	Часть 1, секция 15.2.6	-	te	<p>Неизвестно, что подразумевается под фразой «This part shall have no contents» («Эта часть не должна включать в себя содержимого»). Обозначает ли это, что Zip-файл с заявленным именем должен быть пустым? Или что под объявленным именем должен быть создан файл нулевой длины? Или что-либо еще?</p>	<p>При переводе как замечания, так и разъяснения частично теряется игра слов. Ответ на замечание, представленный на английском языке («<i>The phrase "This part shall have no contents" means that a part shall be created, but it shall have no contents (meaning it would be a zero byte file).</i>») ясно говорит, что цитируемое положение проекта требует создания пустого файла нулевого размера, а не пустого архива.</p> <p>Поскольку речь идет о part в контексте OPC (Open Packaging Convention) то другие формулировки при внимательном чтении возникнуть просто не могли.</p>
10.	Часть 1, секция 2.4	строка 22	te	<p>В данной строке присутствует требование совместимости со стандартом Unicode без уточнения версии. Спецификация XML1.0 опирается на стандарт Unicode 2.0, в то же время в приложении А к части 1 спецификации OOXML упоминается Unicode 4.0. Неочевидно, какая из спецификаций подразумевается здесь.</p>	<p>Возражение противоречит духу и букве стандарта Unicode. Рекомендации по указанию версий детально разработаны Unicode Consortium. Среди прочего, он явным образом предупреждает о нецелесообразности «оверспецифицирования версий»: «<i>Since Unicode is an open standard, it is important not to over-specify the version number.</i>» (см. <a href="http://unicode.org/versions/">http://unicode.org/versions/</a>). Проект ISO/IEC DIS 29500 следует этой рекомендации всюду, где не требуется полная детализация.</p> <p>Проект стандарта 29500 основан на XML 1.0. Следование ему подтверждается, например, явным указанием в строке 23, секции 2.4 Части 1. Декларируется соответствие стандарту Unicode и стандарту ISO/IEC 10646-1 с поддержкой UTF-8 и UTF-16 как требуется XML 1.0.</p> <p>Действующая четвертая редакция спецификация XML 1.0 от 2006-09-29 (см. <a href="http://www.w3.org/TR/REC-xml/">http://www.w3.org/TR/REC-xml/</a>) ссылается на Unicode в общем и Unicode 3.1 (3.0). Unicode 2.0 цитируется в библиографии к W3C спецификации XML 1.0 как и спецификация Unicode 3.1. Аналогично, в информативной (ненормативной) части проекта стандарта 29500 упоминается описание также Unicode 4.0. Это не влияет на нормативную часть и реализацию.</p>

11.	Часть 1, секция 9.1.1	-	te	<p>Необходима точная ссылка на спецификацию ASCII, так как она имеет несколько национальных вариаций.</p> <p><i>Предположительно, ссылка должна быть сделана на стандарт ISO/IEC 646:1983 либо на ANSI X3.4-1986.</i></p>	<p>Замечание не является содержательным, так как в указанной короткой статье дан ПОЛНЫЙ явный перечень всех символов, которые могут быть непосредственно использованы для наименования частей (parts) OPC, а также указано, как могут быть использованы дополнительные символы (через эскейп-последовательности %HH), не являющиеся общим подмножеством национальных вариаций стандартов ISO/IEC 646:1983 или ANSI X3.4-1986.</p>
-----	-----------------------	---	----	---	---

12.	Часть 1, секция 9.1.5	-	te	<p>Данный подраздел, размещенный среди вводных материалов, не сопровождается более детальной спецификацией ОРС в части 2. Предположительно, это ошибка разработчиков.</p> <p><i>Поскольку частичное описание не допускается, оно не должно описываться в части 2.</i></p>	<p>Перевод оригинального замечания выполнен не совсем точно. В оригинале рекомендовалось переместить секцию 9.1.5 Части 1 в Часть 2 и заместить ею некоторую часть спецификации (предположительно раздел 9.1.4 Части 2). Хотя такое решение и возможно, оно нецелесообразно по следующим причинам.</p> <p>Интерливинг является достаточно простой концепцией, и зарекомендовал себя как высокоэффективный способ повышения производительности приложений, используемых при обработке больших файлов архивов и потоков данных (видео, аудио и пр.). В целях упрощения реализации OpenXML в данной версии не требуется реализации интерливинга (что декларируется в Части 1, секции 9.1.5). Соответственно, реализация без поддержки интерливинга может быть полной для текущей версии проекта стандарта 29500.</p> <p>Однако ОЧЕНЬ ВЕРОЯТНО, что интерливинг будет включен в нормативную часть следующей версии Open XML. Соответственно, в целях совместимости «вперед» и для объяснения архитектурных решений ОРС принятых в текущей версии, в текст включено описание интерливинга в необходимом для этого минимальном объеме. Действительно, исключение упоминаний о нем может сделать некоторые требования стандарта неочевидными и спровоцировать разработчиков на создание приложений, требующих большей переработки в будущем, либо (хотя такая возможность и маловероятна) на создание частичных реализаций, которые могут оказаться в будущем несовместимыми с новыми версиями стандарта.</p> <p>Рассмотрение интерливинга в Части 2 с предварительным разъяснением его статуса в текущей версии в Части 1 является примером хорошей практики создания мотивированных стандартов с заботой о совместимости «вперед».</p>
-----	-----------------------	---	----	---	---

13.	Часть 4, введение	страница vii, строка 8	te	Выражение, содержащее слова «fully compatible» и «investment» не имеет смысла в данном контексте.	<p>Перевод замечания является неточным и неполным. В оригинале замечания сказано: «<i>An XML markup cannot be “fully compatible” with an “investment”</i>».</p> <p>Вероятно, автор оригинального замечания не совсем понял следующее предложение, находящееся в строках с 5 по 8 страницы vii Части 4: «<i>The goal is to enable the <b>implementation</b> of the Office Open XML formats by the widest <b>set of tools and platforms</b>, fostering interoperability across <b>office productivity applications and line-of-business systems</b>, as well as to support and strengthen <b>document archival and preservation</b>, all in a way that is fully compatible with the large existing investments in Microsoft Office documents.</i>»</p> <p>Видно, что речь идет о совместимости (сочетаемости, отсутствии противоречий) указанных в абзаце выше выделением сущностей (приложений, платформ и процессов) с инвестициями, сделанными в подготовку и хранение документов в форматах Microsoft Office.</p> <p>Действительно, отсутствие возможности гарантировать корректное преобразование документов в унаследованных бинарных форматах в новый XML-формат (а в некоторых случаях и отсутствие автоматических предупреждений о вносимых конвертацией изменениях) в приложениях, использующих только стандарт ISO 26300 и «старые» стандарты, является важным практическим мотивом для поддержки скорейшего принятия ISO/IEC 29500.</p>
-----	-------------------	------------------------	----	---	---

14.	Часть 4, секция 2.15.1.28	-	te	<p>Утверждается, что защита документа должна обеспечиваться принудительно («shall be enforced»), несколькими фразами позже упоминается, что она может быть игнорирована («may be ignored»).</p>	<p>Автор замечания прочел абзац (со строки 6 до строки 11) секции 2.15.1.28 невнимательно. Дело в том, что цитируемые фрагменты находятся в противопоставляемых частях абзаца. В нем говорится, что элемент documentProtection (Document Editing Restrictions) предназначен только для управления защитой (блокировкой) от непреднамеренного изменения и «добросовестное» приложение должно («shall be enforced») соблюдать ее, если опция блокировки включена. В то же время установка режима блокировки в элементе documentProtection не предполагает каких-либо средств защиты (типа шифрования) от преднамеренного нарушения установленного режима «недобросовестным» приложением и не может рассматриваться как средство обеспечения безопасности (security feature). Соответственно, в последнем предложении абзаца содержится предупреждение, что она может быть проигнорирована («may be ignored»), если такое установлено политикой безопасности приложения (которое может и не быть «добросовестным»).</p> <p>Разумеется, не следует полагаться на тег documentProtection как на надежное средство защиты информации, содержащейся в файле формата Open XML.</p>
-----	---------------------------	---	----	---	--

15.	Часть 4, секция 2.15.1.28	-	te	<p>Описан алгоритм хеширования, предположительно основанный на алгоритме предыдущего поколения, использовавшемся в Word. Этот алгоритм известен слабой устойчивостью, известны опубликованные способы его взлома. Безусловно, следует принять во внимание возражение, что в рамках OOXML устойчивость алгоритма хеширования не всегда существенна, так как пользователь может эффективно снять защиту, попросту распаковав zip-файл и отредактировав XML, чтобы удалить или заменить используемый ключ. Однако некоторые типы программ (например, интерфейс Google Documents), предоставляющие доступ на редактирование и/или чтение документов, способны в то же время обеспечить защиту документа, запретив физический доступ к его содержимому различными способами. С этой точки зрения доступ на редактирование документа не тождественен доступу к его XML-содержимому. Следовательно, имеет смысл применять для защиты документа устойчивый к взлому и в то же время переносимый алгоритм, например, SHA-256.</p>	<p>Авторы замечания, вероятно, не дочитали критикуемый раздел до конца. Они ознакомились только с простым и быстрым алгоритм хеширования, приведенным в стандарте в качестве минимально необходимого. Разработчики стандарта отдают себе отчет в росте угрозы взлома данного алгоритма специалистами, обладающими необходимыми навыками и средствами. В связи с этим они прямо предусмотрели в проекте следующее:</p> <ol style="list-style-type: none"> <li>1. Стандарт явным образом описывает использование следующих сильных алгоритмов (см. список в Части 4 стр.1166): MD2, MD4, MD5, SHA-1, MAC, RIPEMD, RIPEMD-160, HMAC, SHA-256, SHA-384, SHA-512.</li> <li>2. Стандарт описывает возможность расширения - подключение произвольного алгоритма хеширования (атрибут algIdExt (Cryptographic Algorithm Extensibility). Эта возможность необходима как для использования более сильных алгоритмов в связи с ростом вычислительных мощностей потенциальных взломщиков так и для соответствия нормативным требованиям в некоторых юрисдикциях.</li> <li>3. В комментариях разъясняется, как, вероятно, будет развиваться стандарт в дальнейшем в части использования более сложных, чем криптографические хеши, средств.</li> </ol> <p>Ни SHA-256, ни другие сильные алгоритмы, ни стандарт ISO/IEC 10118-3 (cryptographic hash) не упоминаются в спецификации ODF, в связи с чем представляется сомнительным, что ODF может составить базу действительно стандартной и интероперабельной реализации. Некоторые алгоритмы (только MD2, MD5 и SHA-1) упоминаются в цитируемом в ODF источнике: В. Kaliski, PKCS #5: Password-Based Cryptography Specification Version 2.0, IETF, 2000. Из них только SHA-1 удовлетворяет ISO 10118-3.</p> <p>Во всех случаях, когда требуется переносимость и поддержка сильного крипто-хеши можно рекомендовать использование OpenXML.</p>
-----	---------------------------	---	----	--	--

16.	Часть 4, секция 2.15.1.28	строка 13	te	<p>Описание алгоритма не позволяет определить кодировку пароля. Предположительно, это Unicode, однако на практике это может быть Unicode с различным порядком следования байтов: UTF-16BE, UTF16-LE или UTF16 с отметкой порядка байтов.</p> <p>Описанные алгоритмы используют манипуляции с данными на уровне байтов, реализация которых зависит от архитектуры аппаратного обеспечения и поддерживаемого ею порядка следования байтов. Необходимо, чтобы все предположения о порядке следования байтов были четко объявлены.</p>	<p>Авторы замечания не изучили описание критикуемого алгоритма. В строке 18 стр. 1158 части 4 не предположительно, а явно указано, что используются «Unicode characters». В то же время, алгоритм приведен в аппаратно и ОС-независимой форме и к нему не применимы требования выбрать машинно-зависимый порядок байтов в кодировке.</p> <p>UTFBE и UTFLE – описания отображения кодировки на последовательность байтов в машинном слове, а не описания кодировки самой по себе. Как указано выше в разъяснении к замечанию 10 проект стандарта 29500 следует определениям стандарта Unicode и рекомендациям Unicode Consortium с учетом ограничений XML 1.0 где применимо. В описании критикуемого алгоритма нигде не применяются термины “big” и “little” используемые для указания физического размещения байтов в машинно- или ОС-зависимой архитектуре. Используются только термины “low byte” и “high byte” для указания логического порядка UTF-16.</p> <p>Соответственно, разработчик, например в среде GNU/Linux может откомпилировать свое приложение с ключом порядка, соответствующим архитектуре целевой платформы. Более того, авторы разъяснения полагают, что можно выполнить корректную буквальную реализацию алгоритма используя UTF-32. Упомянутая в некоторых родственных замечаниях возможность свести все буквы к одному символу возможна только в случае ошибочного выбора ключа или при ошибке низкоуровневого кодирования строк (неверное предположение о целевой платформе), что не имеет отношения в рассматриваемому проекту.</p> <p>Остается вероятность того, что совпадут хеши по существу разных паролей, набранных в разных языках/кодировках с совпадающими значимыми “low byte” и “high byte”, но ее следует признать чисто теоретической.</p>
-----	---------------------------	-----------	----	--	---



17.	Часть 4, секция 2.15.1.28	стр 1159, строки 6-9	te	<p>Описанные этапы обработки приводят к двусмысленным результатам. Как минимум, результат работы алгоритмов SHR и SHL зависит от архитектуры аппаратного обеспечения, а также от того, являются ли обрабатываемые данные знаковыми или беззнаковыми значениями.</p>	<p>См. разъяснение к замечанию 16.</p> <p>Способ формирования последовательности байтов разъяснен на стр. 1158 Части 4. Все выбираемые из таблицы значения явно представлены в соответствующих частях описания алгоритма (см., например, строку 12 стр. 1159. Все операции описаны явным образом, не зависимым от машинной реализации. Например: "...for every bit in the character, starting with the least significant and progressing to (but excluding) the most significant....".</p>
18.	Часть 4, секция 2.15.1.86	-	te	<p>Элемент использует битовую маску для фильтра отображаемых стилей. Использование битовых масок вместо булевых типов делает практически невозможной работу со стандартными технологиями обработки XML, наподобие XSLT, которые не поддерживают работу с операциями битового уровня.</p>	<p>Использование битовых масок для определения отображаемых стилей (включая описания таблиц и подстановки {субрегионов} шрифтов) не противоречит спецификации XML и не ограничивает возможности валидации. Что требуется в действительности – их полное и однозначное описание в стандарте.</p> <p>Тем не менее, учитывая относительную «ненаглядность» данного средства, члены рабочей группы Еста ТС45 ограничили использование битовых масок только теми случаями, где выигрыш очевиден, а область влияния ограничена и/или где это вытекает из рекомендаций стандартов ISO:</p> <ul style="list-style-type: none"> <li>• таблицы стилей (Секции 2.4.51, 2.4.52, 2.3.1.8, 2.4.7, 2.4.8 и 6.1.2.7);</li> <li>• подстановки стилей для отображения документа в пользовательском интерфейсе, обычно отличном от целевой платформы рендеринга (секции 2.15.1.86 и 2.15.1.87);</li> <li>• спецификации подстановок языков, субрегионов, шрифтов (секция 2.8.2.16). Здесь 6 масочных атрибутов заменяют 96 тегов, которые пришлось бы использовать при ином подходе.</li> </ul> <p>Значения битовых масок font/sig полностью соответствуют значениям, определенным в ISO 14496-22. Отсутствие полной поддержки этого стандарта в ODF может ограничить его применение в некоторых регионах/с некоторыми шрифтами.</p>

19.	Часть 4, секция 2.15.1.86	-	te	<p>Элемент использует битовую маску для параметров сортировки отображаемых стилей. Использование битовых масок вместо булевых типов делает практически невозможной работу со стандартными технологиями обработки XML, наподобие XSLT, которые не поддерживают работу с операциями битового уровня.</p>	<p>См. разъяснение к п.18. Битовые маски позволяют оптимизировать сравнение значений. Корректно определенные битовые маски не накладывают ограничений на использование стандартных технологий обработки XML и могут быть обработаны с использованием XSLT.</p> <p>Так, например, используя W3C Schematron (XML structure validation language) можно верифицировать/валидировать файлы с полями битовых масок (они интерпретируются как бинарные строки или HEX). См. также ISO/IEC 19757 - Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation – Schematron.</p> <p>На практике можно использовать существующие программные продукты независимых поставщиков, такие как Altova XML Spy 2007 для валидации и верификации документов OpenXML.</p>
20.	Часть 4, секция 2.15.2.32	-	te	<p>Игнорируется существование браузеров, отличных от Internet Explorer. Ни Safari, ни Opera, ни Firefox не могут быть обозначены в качестве целевого браузера.</p> <p>Секция требует, чтобы «все настройки, не совместимые с целевым браузером, были деактивированы». Это, предположительно, исключает ситуацию генерации вывода в форматах, соответствующих стандартам, но не являющихся рекомендованными для Internet Explorer (например, использование PNG, MathML, SVG вместо VML).</p>	<p>Тег optimizeForBrowser не указывает конкретный целевой браузер, а только совокупность свойств. Некоторым совокупностям ограничительных свойств отвечают поименованные для определенности версии Microsoft Internet Explorer и Netscape Navigator.</p> <p>В рамках стандарта форматов офисных документов не предполагается требовать от приложений дополнительной реализации таких веб-ориентированных технологий, как MathML или SVG, поскольку в них нет необходимости для описания формата документа.</p> <p>В то же время, стандарт, очевидно, не запрещает приложениям реализацию дополнительных спецификаций, и приложение, поддерживающее SVG или MathML, например OpenOffice.org, может их использовать для оптимизации вывода в HTML на основе информации тега optimizeForBrowser.</p> <p>Что касается PNG, то для этого формата элемент как раз предусмотрен: allowPNG.</p>

21.	Часть 4, секция 2.15.3	-	te	<p>Настройки «совместимости» не решают ни одной общей проблемы. Они фактически представляют собой коллекцию форматов настроек предыдущих поколений Microsoft Word. При этом не предоставляется аналогичных возможностей для поддержки форматов настроек предыдущих поколений от других приложений. Возможным вариантом решения могла бы быть поддержка специфичных для приложения настроек через существующие механизмы расширения OOXML без специального включения их в стандарт.</p>	<p>«Общее» решение проблемы совместимости, предлагаемое авторами замечания, разумеется разрешено в OpenXML (как и в ODF). Но именно оно не решает никаких конкретных проблем.</p> <p>Параметры совместимости являются необязательными к указанию и используются для поддержки визуальной точности воспроизведения документов, созданных в предыдущих версиях программного обеспечения. Обрабатывающие документ приложения могут игнорировать эти параметры и нет необходимости добавлять их, за исключением случаев, когда это необходимо для обратной совместимости.</p> <p>Более того, допустимые значения параметров не привязаны к предыдущим версиям Microsoft Word, а являются общими для большинства текстовых процессоров. В частности, практически все особенности поведения предыдущих версий WordPerfect могут быть описаны через данный механизм. Также, во многих случаях старые продукты Microsoft «имитировали» поведение старых версий офисных продуктов Lotus, а современный OpenOffice.org имитирует поведение старых версий Microsoft Office. Все эти случаи могут быть охвачены перечисленными настройками.</p>
-----	------------------------	---	----	--	---

22.	Часть 4, секция 2.15.3.26	-	te	<p>Элемент «footnoteLayoutLikeWW8» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.</p>	<p>Указанный параметр включен в стандарт, поскольку его применение облегчает воспроизведение точного вида документов, конвертируемых из бинарных форматов и подготовленных в унаследованных приложениях (и известно, что таких документов имеется значительное множество, особенно в государственных организациях). Наличие этого параметра в документе указывает, что в приложении, из которого был сохранен документ, было использовано соответствующее оформление. Другое приложение, которое может быть использовано для дальнейшей работы с документом, может применить то же оформление, либо проигнорировать этот параметр, возможно, уведомляя пользователя об этом.</p> <p>В описании проекта стандарта 29500 подробно документировано значительное число допустимых параметров (которых порядка 200 только для WordprocessingML), и лишь небольшая часть параметров не описана однозначно по причинам, которые каждый раз разъяснены.</p> <p>Забавно, но данное замечание просто переписывает предупреждение авторов проекта стандарта Open XML (стр.1416 Части 4): <i>"[Guidance: To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated WordprocessingML due to issues with its output, and is maintained only for compatibility with existing documents from that application. end guidance] Typically, applications shall not perform this compatibility.»</i></p> <p>Как подробно показано, например, в <a href="http://blogs.msdn.com/brian_jones/archive/2007/01/09/specifying-the-document-settings.aspx">http://blogs.msdn.com/brian_jones/archive/2007/01/09/specifying-the-document-settings.aspx</a> проблема описания настроек совместимости является общей для проекта 29500 (Open XML) и стандарта 26300 (ODF). Однако ODF вообще практически не дает описаний настроек совместимости, что делает достижение реальной интероперабельности трудно достижимым.</p>
-----	---------------------------	---	----	---	--

23.	Часть 4, секция 2.15.3.31	-	te	Элемент «lineWrapLikeWord6» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	<p>См. разъяснение по п.22.</p> <p>Цель, которая была поставлена при разработке стандарта Open XML, заключалась в поддержке максимальной совместимости с существующими документами. В данном конкретном случае, описание функциональности в стандарте приведено для лучшего взаимодействия приложений разных производителей.</p> <p>Поскольку обсуждаемый параметр предназначен для реализации функциональности устаревших приложений, он не является обязательным к реализации в каждом приложении, поддерживающем стандарт Open XML.</p> <p>Однако, если приложение уже поддерживает подобную функциональность, то спецификация предлагает способ чтения и записи этого параметра. Если же подобной функциональности в приложении, работающем с документом, не предусмотрено, то параметр может быть проигнорирован.</p> <p>Полезно напомнить ситуацию с OpenOffice.org, который, как всякое развитое офисное приложение, поддерживает и вставляет в документы значительное число опций совместимости, но ODF не дает ему достаточных выразительных средств, чтобы предупредить об этом третьи приложения. OpenOffice.org улучшит управление совместимостью благодаря поддержке OpenXML.</p>
24.	Часть 4, секция 2.15.3.32	-	te	Элемент «mwSmallCaps» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	Комментарий аналогичен приведенному в п. 23.
25.	Часть 4, секция 2.15.3.41	-	te	Элемент «shapeLayoutLikeWW8» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	Комментарий аналогичен приведенному в п. 23.

26.	Часть 4, секция 2.15.3.51	-	te	Элемент «suppressTopSpacing» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	Комментарий аналогичен приведенному в п. 23.
27.	Часть 4, секция 2.15.3.54	-	te	Элемент «uiCompat97To2003» определен как «Отключение функциональности пользовательского интерфейса, несовместимой с Word97-2003». Не уточняется желаемое поведение программ, не являющихся Microsoft Word. Также не поддерживается аналогичная возможность для отключения настроек, не совместимых с определенными версиями других распространенных программ семейства офисного ПО (наподобие OpenOffice или WordPerfect Office) – например, отключение настроек, не совместимых с OpenOffice 1.5 или WordPerfect 8. Не предполагается универсальной возможности для независимых разработчиков указывать аналогичные параметры для своего ПО.	<p>В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.</p> <p>Если ознакомиться с Частью 4 проекта стандарта внимательно, то можно найти, что он достаточно подробно описывает управление настройками для имитации поведения Word Perfect.</p> <p>Управление настройками совместимости для OpenOffice.org 1.5 в проект стандарта действительно не включено. Можно предположить, что члены комитета ECMA-45 исходили при этом из следующих соображений:</p> <ul style="list-style-type: none"> <li>• Число подготовленных в нем документов исчезающе мало (возможно тогда многие другие продукты, например, Лексикон, тоже должны быть явно описаны);</li> <li>• Одной из целей создания ранних версий OpenOffice.org была имитация поведения еще более ранних версий Microsoft Office. Соответственно, разработчики могут использовать предназначенные непосредственно для этого теги.</li> </ul>

28.	Часть 4, секция 2.15.3.54	-	te	Элемент «truncateFontHeightsLikeWP6» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	<p>Для данного элемента дано определение, примеры, включая примеры вывода, оговорены ограничения его использования, и разъяснено почему нельзя включить полное описание соответствующего поведения (техническую спецификацию на WordPerfect 6) в стандарт. Наиболее вероятно, что применяться оно будет только с приложениями, имеющими функциональность, необходимую для конвертации файлов из WordPerfect 6. Альтернативное решение (например, использование «чистого» ODF) оставило бы соответствующий корпус документов бинарным либо привело бы к неуправляемым изменениям представления документов (возможно, без предупреждений).</p> <p>В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.</p>
29.	Часть 4, секция 2.15.3.6	-	te	Элемент «autoSpaceLikeWord95» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	<p>В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.</p> <p>Аналогично ответу на замечание 28.</p>
30.	Часть 4, секция 2.15.3.63	-	te	Элемент «useWord2002TableStyleRules» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	<p>В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.</p>
31.	Часть 4, секция 2.15.3.64	-	te	Элемент «useWord97LineBreakRules» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	<p>В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.</p>

32.	Часть 4, секция 2.15.3.65	-	te	Элемент «wpJustification» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек.
33.	Часть 4, секция 2.15.3.66	-	te	Элемент «wpSpaceWidth» определен в терминах поведения приложения предыдущего поколения. Стандарт содержит недостаточно деталей, описывающих, как воспроизвести это поведение.	В секции 2.15.3 описаны опциональные настройки, позволяющие обеспечить совместимость с предыдущими версиями Microsoft Word. От приложений, поддерживающих OOXML не требуется обеспечения поддержки этих настроек
34.	Часть 4, секция 2.16.5.33	-	te	Отсутствует описание способа именования изображений. Не уточняется, должен ли это быть URI, или путь в локальной файловой системе, или что-то еще. Приведенный пример содержит файловый путь для DOS, и, как следствие, не является переносимым.	Данный комментарий является следствием неверного толкования спецификации. Секция 2.16.6.33 представляет собой всего лишь пример. Имеется возможность извлечения изображения из документа по имени или по URL.
35.	Часть 4, секция 2.16.5.33	-	te	В подклассе определяется поле INCLUDEPICTURE, которое «получает изображение, содержащееся в именованном документе». При этом нигде не упоминается, какие форматы допустимы для изображения.	Поле INCLUDEPICTURE используется для получения изображений из документов. Некоторые из допустимых форматов изображений описаны в секции 5.2.13 стандарта Ecma-376.  Примерами форматов изображений являются JPG, PNG, BMP и др. Полный перечень поддерживаемых форматов зависит от поддерживаемых приложением и операционной системой «графических фильтров». Нецелесообразно ограничивать его фиксированным списком в данном проекте стандарта.
36.	Часть 4, секция 2.16.5.34	-	te	Отсутствует описание способа именования документов. Не уточняется, должен ли это быть URI, или путь в локальной файловой системе, или что-то еще. Приведенный пример содержит файловый путь для DOS, и, как следствие, не является переносимым.	Данный комментарий является следствием неверного толкования спецификации. Имеется возможность задания имени документа как в формате локальной файловой системы, так и в формате URI.  См. разъяснение к возражению 37.



37.	Часть 4, секция 2.16.5.34	-	te	<p>В подклассе определяется поле INCLUDETEXT, которое «вставляет полностью или частично текст и графику, содержащиеся в именованном документе». При этом нигде не упоминается, какие форматы допустимы для получаемого текста.</p>	<p>INCLUDETEXT включает текст и графику из внешнего источника в текущий документ. Операция по включению текста аналогична операции импорта из внешнего источника в каком либо альтернативном формате и поэтому нет необходимости давать ограниченный перечень форматов из которых разрешено включение текста. Тем не менее, перечень основных форматов дан в Секции 11.2.3, на странице 28. Указаны следующие форматы:</p> <ul style="list-style-type: none"> <li>• Text = application/txt</li> <li>• RTF = application/rtf</li> <li>• HTML = application/html</li> <li>• XML = application/xml</li> </ul> <p>Как указано в описании INCLUDETEXT в части 4, секции 2.16.5.34 на странице 1538 – Текст из другого WordprocessingML документа, также может быть включен: <i>«If the document is a WordprocessingML document, the portion marked by the optional bookmark field-argument-2 is inserted.»</i></p>
38.	Часть 4, секция 2.16.5.34	-	te	<p>Флаг \t позволяет применить именованную XSLT-трансформацию для XML-файла на вводе и вставить результаты в вывод. При этом нам остается неизвестно, какую конкретную версию XSLT-трансформации разрешено использовать.</p>	<p>Данное замечание относится к функции INCLUDETEXT. В случае вставки текста из XML возможно указать XSLT преобразование, которое будет применяться к включаемому тексту. Данное действие близко к конвертации данных из внешних источников в текущий документ и позволяет различным приложениям реализовывать его различным образом, т.к. результат вставки будет отформатирован как WordprocessingML.</p> <p>С другой стороны версия XSLT не определяется, т.к. данный флаг определяет xsl файл, с описанием XSLT преобразования. Версия XSLT будет указана в данном файле. Приложение может интерпретировать различные версии XSLT.</p>

39.	Часть 4, секция 2.16.5.41	-	te	<p>Описано поле “MACROBUTTON”, которое может запускать определенный макрос или команду. Однако отсутствует указание, какой язык программирования или API возможно использовать для данного макроса или команды.</p>	<p>Поле MACROBUTTON позволяет запустить макрос. OOXML разрешает исполнять макросы при определенных условиях в определенных типах файлов, однако OOXML не определяет языки программирования или API для макросов. Данный вопрос находится за пределами спецификации и не должен быть описан в ней.</p> <p>Как указано в 2.18.58, страница 1760, под заголовком «ST_MacroName (Script Subroutine Name Value)»: «<i>The language and location of this subroutine may be determined using any method desired by an application.</i>». Это же высказывание применимо и в данном случае. Использование имени макроса в поле MACROBUTTON не накладывает ограничений на возможные реализации макроязыков.</p>
40.	Часть 4, секция 2.16.5.41	Стр. 1512, строки 11-12	te	<p>Судя по тексту, поле AUTONUM не рекомендуется к использованию. Новый стандарт не должен содержать нерекондуемых частей.</p>	<p>Поле AUTONUM во вновь создаваемых документах рекомендуется заменять на LISTNUM. Однако, OOXML создавался для обеспечения максимальной совместимости с уже существующими массивами документов, поэтому, поддержка разметки, принятой в предыдущих версиях документов является существенной. В связи с чем использование данного поля возможно, но не рекомендуется для новых документов.</p> <p>Практика включения в стандарты не рекомендованных но существующих объектов достаточно распространена. Она особенно характерна для стандартов, являющихся результатом длительных теоретических изысканий и обобщением практики. В полной мере ее может избежать только очень ограниченный стандарт с очень узкой областью применимости в очень консервативной сфере.</p>

41.	Часть 4, секция 2.18.4	-	te	<p>Представленные иллюстрации имеют низкое качество, не предоставляющее ни желаемого масштабирования, ни расстояния между элементами, ни глубины цвета и т.д. Небольшая диаграмма, приведенная в качестве примера, недостаточно описана. Например, неясно, являются ли размеры границ абсолютными либо они масштабируются вместе с размером листа. Также некоторые изображения, например, 'apples' или 'balloons3Colors', содержат артефакты копирования, такие как наружные линии или пятна.</p>	<p>Замечание относится к описанию в тексте стандартного типа ST_Border. Данный тип описывает возможный вид границ для объектов, которые могут иметь границы. И содержит повторяющиеся изображения, которые могут быть использованы в качестве границ вокруг объектов. Размеры границ определены в абсолютных величинах и в документе приведены в увеличенном виде, что вызвало искажения.</p> <p>Если национальный орган стандартизации сочтет это необходимым, он может перекалифицировать замечание в «ге» (редакционное) и рекомендовать при подготовке официального издания стандарта откорректировать качество линий и иллюстраций.</p>
42.	Часть 4, секция 2.18.4	-	te	<p>Отсутствует механизм для расширения предоставленного набора обрамления границ. Поскольку указанные обрамления границ являются сильно ориентированными на Запад, то было бы хорошо предоставить возможность приложениям поддерживать графические стили, специфичные для конкретного региона.</p>	<p>В приведенной секции (описание типа ST_Border) указаны изображения, которые могут применяться для оформления границ объектов с границами.</p> <p>Кроме достаточно большого набора линий, есть, также, набор изображений и орнаментов.</p> <p>Например, изображения – соски-пустышки, воздушного шарика, воздушного шара с гондолой, яблока, дома, бутылки с шипучим напитком, елки, божьей коровки, глобуса с изображением Африки, части Азии и Европы, кленового листа, пальмы, карандаша..., а также большое число орнаментов. Утверждение о сильной «ориентации на Запад» данного перечня картинок представляется излишне преувеличенным.</p> <p>Более того, данный тип определяет изображения, которые уже де-факто используются в существующих документах, что позволяет обеспечить их однозначный перенос в новый стандарт.</p>

43.	Часть 4, секция 2.18.45	-	te	<p>Указано, что длина должна соответствовать «ровно 3 символам». Это противоречит представленному примеру, в котором длина равна 6 символам.</p>	<p>В данном случае, возможно, произошло недопонимание при прочтении секции, где описан тип данных для определения цвета в RGB координатах. Цвет формируется как 3 байта RGB с значениями от 0 до 255, при этом записываются эти значения в шестнадцатеричной форме, т.е. цвет формируется как 3 октета, что при шестнадцатеричной записи будет составлять 6 символов. В ограничении указан тип <code>base="xsd:hexBinary"</code>, который определяет размер строки в октетах. Данный способ определения цвета общепринят и используется, например в HTML.</p>
44.	Часть 4, секция 2.18.51	-	te	<p>Использование перечисления из 255 языковых кодов в дополнение к набору кодов ISO-639-1 не предоставляет никакой выгоды, одновременно увеличивая программные затраты на обработку документа в формате OOXML.</p>	<p>Замечание относится к базовому типу <code>ST_Lang</code>, секция 2.18.51, которая описывает альтернативы указания языка. Это могут быть шестнадцатеричный код языка, либо ISO 639-1 строка, символ дефиса, ISO 3166-1 alpha 2 строка. Т.е. стоит отметить, что на ряду с нумерацией языков, стандарт предусматривает явное указание строковых наименований языков принятых в соответствующих ISO стандартах. Двухбайтовое указание кода языка предусмотренное типом <code>ST_LangCode</code> (секция 2.18.52) не усложняет, а упрощает обработку документов и позволяет упростить конвертацию уже существующих документов, где принят подобный способ кодирования языка.</p>

45.	Часть 4, секция 2.18.66	-	te	<p>Описанная система форматирования не является исчерпывающей. В частности, не поддерживаются ряд используемых по настоящее время систем счисления (Армянская, Тамильская, Эфиопская, Кхмерская, система счисления на основе греческого алфавита), а также ряд систем счисления исторического характера, используемых в образовании.</p>	<p>Можно безусловно согласиться с тем, что введенная система нумерации не является исчерпывающей. В стандарте введено 60 способов нумерации, имеющих достаточно большое распространение в мире, включая русскую, хинди, китайские, корейские, японские, арабские и пр. Проблема формирования более полного списка состоит в том, что ряд национальных традиций имеет неоднозначные или не вполне определенные способы нумерации.</p> <p>ЕСМА-45 проделала огромную работу в этой области, суммировав для первой версии стандарта исключительно широкий выбор возможностей. В настоящее время это наиболее полный стандартный перечень нумераторов.</p> <p>Достаточно сравнить предложенное в OpenXML решение с тем, что имеется в ODF (ISO/IEC 26300). Вот фрагмент из спецификации ODF:</p> <p><b>12.2.2 Format Specification</b></p> <p>The <code>style:num-format</code> attribute specifies the format of the number in the same way as the [XSLT] format attribute. The number styles supported are as follows:</p> <ul style="list-style-type: none"> <li>• Numeric: 1, 2, 3, ...</li> <li>• Alphabetic: a, b, c, ... or A, B, C, ...</li> <li>• Roman: i, ii, iii, iv, ... or I, II, III, IV,...</li> </ul> <p>The value of this attribute can be "1", "a", "A", "i", or "I". For some elements, the attribute value also can be empty. In this case, no number is displayed.</p> <p>Видно, что в ODF не поддерживается не только Армянская, Тамильская, Эфиопская, Кхмерская, но даже и кирилличная нумерация. Разумеется, и OpenXML и ODF позволяют создать аналоги нумерованных списков иными способами, однако при этом теряются некоторые достоинства структурированного представления текстов.</p> <p>Национальный орган стандартизации, считающий, что необходимо включить дополнительный способ нумерации списков в следующую версию стандарта, может внести соответствующее предложение (что обычно требует довольно тщательного проектирования. Например, нужно ли в русском нумераторе использовать буквы «й» (и краткое), «ё», твердый и мягкий знаки и пр.? В действительности ответ был определен уже давно – не нужно).</p>
-----	-------------------------	---	----	--	---

46.	Часть 4, секция 2.18.66	-	te	<p>Данная секция не предоставляет никаких нормативных определений, за исключением нескольких перечисляемых значений. Также не предоставлено никакого нормативного истолкования этих значений.</p> <p>Приведенный информативный пример недостаточен для любых ситуаций, за исключением полностью тривиальных случаев. Например, неясно, где определяется «Korean Legal Counting System».</p>	<p>Список типов нумераторов перечисляет общепринятые способы нумерации (как правило, совпадающие с соответствующими наборами цифр или букв и уже определенными в Unicode), и не ставит своей целью дать их новые описания.</p>
47.	Часть 4, секция 2.18.66	«chicago»	te	<p>Определение формата ссылается на «Chicago Manual of Style» в целом, без указания редакции и страницы данного документа.</p>	<p>На практике нет необходимости ссылаться на номер издания или страницу «Chicago Manual of Style». Дело в том, что требуемые для определения нумераторов разделы не зависят от издания, не претерпевают изменений и, вероятно, не претерпят в дальнейшем.</p> <p>Также доступно официальное электронное издание:  <a href="http://www.chicagomanualofstyle.org/home.html">http://www.chicagomanualofstyle.org/home.html</a></p> <p>Данное справочное руководство хорошо известно отечественным техническим редакторам, но, в случае, если национальный орган стандартизации считает это полезным, он может переквалифицировать это замечание в редакционное («re») и представить в JTC-1.</p>
48.	Часть 4, секция 2.18.66	«decimalFull Width» и т.д.	te	<p>Присутствует несколько упоминаний двухбайтной и однобайтной форм записи арабской системы счисления. Учитывая, что условием совместимости с OOXML является использование Unicode-кодировок UTF8 или UTF16, упоминание других кодировок в тексте противоречит данному требованию.</p>	<p>Замечание основано на недоразумении. Данное описание имеет отношение к внешнему виду символов при использовании нумераторов, а не к форме их хранения в документе.</p> <p>Вариант DecimalFullWidth сохранен для совместимости с существующими документами.</p>

49.	Часть 4, секция 2.18.66	«lowerLetter» и т.д.	te	Определяется, что в некоторых системах счисления используются алфавитные символы, но не описывается способ записи чисел, превышающих число букв в алфавите.	Данное поведение достаточно полно определено в разделе 2.16.4.3 где описаны функции преобразования чисел в алфавитные символы. Предполагается использование описанных функций для получения изображения в виде символов, соответствующего очередному номеру в нумерованном списке. В качестве примера алфавитного преобразования приводится число 54, при преобразовании в латинские символы, будет получена строка «zz».
50.	Часть 4, секция 2.18.66	«numberInDash» и т.д.	te	Определяется использование знаков «черты» («dash»), окружающих число, но не уточняется, какой из Unicode-символов этого семейства имеется в виду (возможные варианты: en-dash, em-dash, figure-minus, hyphen-dash, quotation-dash и др.)	Речь идет о стиле нумерации вида: « - 1 - »  Разумеется, в этом месте и всюду в аналогичном контексте имеется в виду символ hyphen-minus.  Однако возможно, следует в окончательной редакции определить это прямо.
51.	Часть 4, секция 2.18.72	-	te	Отсутствует определение «Классификации Panose-1» («Panose-1 classification») для шрифта.	Данный комментарий, возможно, основан на неполном прочтении необходимых разделов спецификации OOXML. Часть 4, секция 2.18.72. использует классификацию Panose-1 в элементе panose1 (Panose-1 Typeface Classification Number), определенном в секции 2.8.2.13, которая имеет прямую отсылку на определение в PANOSE Classification Guide, Version 1.2. Общая ссылка на этот Guide помещена также в библиографии к части 1. Отметим, что это же определение обычно используется при формировании метаданных шрифтов их поставщиками.  Интересно отметить, что в ODF 1.0 panose явно упоминается только в использовании атрибутов (стр. 478, <attribute name="svg:panose-1"/>) и никак не определяется. Возможно, необходимо использовать косвенные ссылки через спецификацию SVG.

52.	Часть 4, секция 2.18.85	-	te	Нехватка определений шаблонов заполнения. Предоставленные иллюстрации недостаточны. Для реализации приложений необходимо знать, какая степень соответствия иллюстрации является обязательной.	В определениях достаточно четко указан способ образования шаблона. Возможно, стоит привести изображения заливок с более высоким качеством в финальной версии документа.
53.	Часть 4, секция 2.3.1.8	-	te	Элемент использует битовую маску для определения различных свойств условного форматирования абзаца. Использование битовых масок вместо булевых типов делает практически невозможной работу со стандартными технологиями обработки XML наподобие XSLT, которые не поддерживают работу с операциями битового уровня.	<p>См. разъяснения к замечаниям 18, 19.</p> <p>Битовые маски использованы для обеспечения возможности формульно вычислять значения форматирования. Это обеспечивает более простой и гибкий способ вычисления параметров форматирования для иерархий объектов, что в конечном итоге позволяет сделать реализации формата более эффективными и менее требовательными к вычислительным ресурсам. Битовые маски могут иметь некоторое использование и при XSLT преобразованиях.</p> <p>Также, данная возможность повышает степень совместимости с существующим массивом документов.</p> <p>Как указано в разъяснении к замечанию 19, корректно определенные битовые маски не накладывают ограничений на использование стандартных технологий обработки XML и могут быть обработаны с использованием XSLT.</p> <p>Так, например, используя W3C Schematron (XML structure validation language) можно верифицировать/ валидировать файлы с полями битовых масок (они интерпретируются как бинарные строки или HEX). См. также ISO/IEC 19757 - Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation – Schematron.</p>



54.	Часть 4, секция 2.3.1.19	-	te	<p>Утверждается, что «свойства расположения данного встроенного объекта описываются с помощью VML-синтаксиса». Это противоречит заявленному в Части 1 секции 8.2.6 утверждению, что «VML должен рассматриваться как устаревший формат, включенный в стандарт исключительно в целях совместимости с приложениями предыдущего поколения. В новых приложениях для описания чертежей и рисунков настоятельно рекомендуется использовать DrawingML». Отсюда следует, что в новых документах при встраивании OLE-объектов не должен использоваться синтаксис VML. В противном случае всем пользователям OOXML придется поддерживать VML, даже если работа с документами предыдущего поколения не предполагается.</p>	<p>Ошибка перевода. Секция 2.3.1.19 не упоминает VML. В действительности замечание относится к секции 2.3.3.19.</p> <p>Речь идет о способе вставки изображений внедренных объектов, включенных в документ. Сами объекты хранятся в отдельной папке в OPC пакете, однако их изображения отображаются в документе. Для указания положения изображений объектов используется vml. Это – простейший и самый быстрый способ не связанный с описанием сложных графических объектов, для которых рекомендуется DrawingML. Данное решение позволяет также добиться совместимости с существующими массивами документов.</p> <p>Возможно, в следующих версиях стандарта будет также введена поддержка позиционирования внедренных объектов средствами DrawingML, однако в настоящей версии комитет ECMA-45 признал это преждевременным.</p>
-----	--------------------------	---	----	--	---

55.	Часть 4, секция 2.4.51	-	te	<p>Элемент использует битовую маску для определения различных свойств форматирования таблиц. Использование битовых масок вместо булевых типов делает практически невозможной работу со стандартными технологиями обработки XML наподобие XSLT, которые не поддерживают работу с операциями битового уровня.</p>	<p>См. разъяснения к замечаниям 18, 19, 53.</p> <p>Битовые маски использованы для обеспечения возможности формульно вычислять значения форматирования. Это обеспечивает более простой и гибкий способ вычисления параметров форматирования для иерархий объектов, что в конечном итоге позволяет сделать реализации формата более эффективными и менее требовательными к вычислительным ресурсам. Битовые маски могут иметь некоторое использование и при XSLT преобразованиях.</p> <p>Также, данная возможность повышает степень совместимости с существующим массивом документов.</p> <p>Как указано в разъяснении к замечанию 19, корректно определенные битовые маски не накладывают ограничений на использование стандартных технологий обработки XML и могут быть обработаны с использованием XSLT.</p>
-----	------------------------	---	----	---	---

56.	Часть 4, секция 2.4.52	-	te	<p>Элемент использует битовую маску для определения различных исключений при форматировании таблиц. Использование битовых масок вместо булевых типов делает практически невозможной работу со стандартными технологиями обработки XML наподобие XSLT, которые не поддерживают работу с операциями битового уровня.</p>	<p>См. разъяснения к замечаниям 18, 19, 53.</p> <p>В контексте данного замечания речь идет о возможности существенно ускорить форматирование таблиц с применением битовых масок.</p> <p>Битовые маски использованы для обеспечения возможности формульно вычислять значения форматирования. Это обеспечивает более простой и гибкий способ вычисления параметров форматирования для иерархий объектов, что в конечном итоге позволяет сделать реализации формата более эффективными и менее требовательными к вычислительным ресурсам. Битовые маски могут иметь некоторое использование и при XSLT преобразованиях.</p> <p>Также, данная возможность повышает степень совместимости с существующим массивом документов.</p> <p>Как указано в разъяснении к замечанию 19, корректно определенные битовые маски не накладывают ограничений на использование стандартных технологий обработки XML и могут быть обработаны с использованием XSLT.</p>
-----	------------------------	---	----	--	--

57.	Часть 4, секция 3.17.4.1	-	te	<p>Ограничение, подразумевающее использование только двух базисов для дат, является необоснованной привязкой к программной реализации конкретного производителя.</p> <p>Существуют другие возможные базисы для дат, в частности, для представления исторических дат, предшествующих 1900 году.</p>	<p>Данное представление даты закрепляет совместимость с де-факто существующими способами представления даты в большом количестве существующих редакторов электронных таблиц и имеет цель обеспечить совместимость и документированность сложившейся практики работы с электронными таблицами, не предлагая, в настоящий момент изменений, в силу неоднозначностей, которые могут возникнуть у миллионов пользователей в случае перехода на новую систему работы с датами.</p> <p>К сожалению, не представляется возможным исключить данное поведение прямо сейчас (как и ранее) по причинам требований совместимости документов (во многих юрисдикциях электронные таблицы могут выступать непосредственно в роли первичных финансовых документов, приниматься к рассмотрению в суде и пр.). В связи с этим финансовые организации возражают против новелл в данной области.</p> <p>Возможно, в следующих версиях формата, имеет смысл предложить дополнительные представления дат и другие правила работы с ними.</p> <p>См. также разъяснение к замечанию 58.</p>
-----	--------------------------	---	----	--	---

58.	Часть 4, секция 3.17.4.1	-	te	<p>Предписание неправильного расчета дат с использованием 1900 года в качестве базиса является неприемлемым. Стандарт ISO не может предписывать использование некорректных значений для григорианского календаря. Это приведет к путанице, снижению переносимости программ и неиссякаемому потоку ошибок.</p>	<p>Ирония данного замечания состоит в том, что неверное предположение о том, что 1900 г. является високосным, было сделано в табличном процессоре Lotus 1-2-3, ныне принадлежащем IBM. Этот продукт был лидером рынка той эпохи, когда про Microsoft Excel еще никто не слышал. С тех пор все коммерческие табличные процессоры следуют этому неверному соглашению.</p> <p>Данное решение неоднократно обсуждалось ранее. Первоначальные разработчики, а за ними и многие другие, обычно оправдывали его тем, что это было полезно для повышения эффективности, и что действительно затронутым может оказаться намного менее, скажем, десятитысячной доли процента документов (оценка приближительна). Позиция Microsoft изложена, в частности, в сообщении «Excel 2000 incorrectly assumes that the year 1900 is a leap year» (см. <a href="http://support.microsoft.com/kb/214326/en-us">http://support.microsoft.com/kb/214326/en-us</a>).</p> <p>Данное представление даты закрепляет совместимость с де-факто существующими способами представления даты в большом количестве существующих редакторов электронных таблиц и имеет цель обеспечить совместимость и документированность сложившейся практики работы с электронными таблицами, не предлагая, в настоящий момент изменений, в силу неоднозначностей, которые могут возникнуть у миллионов пользователей в случае перехода на новую систему работы с датами.</p> <p>Мнение о неиссякаемом потоке ошибок, кажется надуманным, т.к. данная система нумерации дат фактически используется подавляющим количеством пользователей электронных таблиц во всем мире и возможные ошибки уже стали частью стандартного поведения систем. Изменение данного поведения без отдельного глубокого переосмысления воздействия на огромное количество пользователей кажется необоснованным.</p> <p>Ведущие финансовые институты, участвовавшие в разработке ECMA-376 выразили сомнения в возможности изменения данного поведения в близком будущем.</p>
-----	--------------------------	---	----	---	---

59.	Часть 4, секция 3.17.4.1	стр. 2522, строки 14-18	te	Предлагается система дат, основанная на двух базах. Необходимость иметь две слегка отличающихся системы исчисления дат неочевидна. Кроме того, это приведет к дополнительным затратам по решению связанных проблем, так как потребуются определять систему исчисления, используемую по умолчанию, поддерживать конвертацию между двумя системами и т.п.	Данное представление даты закрепляет совместимость с де-факто существующими способами представления даты в большом количестве существующих редакторов электронных таблиц и имеет цель обеспечить совместимость и документированность сложившейся практики работы с электронными таблицами, не предлагая, в настоящий момент изменений, в силу неоднозначностей, которые могут возникнуть у миллионов пользователей в случае перехода на новую систему работы с датами.
60.	Часть 4, секция 3.17.4.1	стр. 2522, строки 16-18	te	Максимальное значение даты приведено как 9999-12-31 00:00:00, что скорее всего не соответствует значению, подразумеваемому предложенной системой исчисления дат. Ожидаемое максимальное значение, исходя из предложенной системы исчисления дат, будет соответствовать 9999-12-31 23:59:59	<p>Формат определяет отдельно представление даты ( в секции 3.17.4.1) и представление времени (в секции 3.17.4.2). Далее в секции 3.17.4.3 описывается, как эти представления могут быть скомбинированы. В частности говорится: <i>«Любой компонент даты может быть добавлен к любому компоненту времени для получения порядкового значения комбинации даты и времени.»</i></p> <p>Т.е. верхний предел ДАТЫ может быть представлен как 9999-12-31 00:00:00 или 9999-12-31 23:59:59 или как либо еще. Верхний предел даты и времени должен быть представлен как 9999-12-31 23:59:59</p>
61.	Часть 4, секция 3.17.4.1	стр. 2522, строки 19	te	Предложенная система не поддерживает даты, предшествующие 1 января 1900 года.	Данное представление даты закрепляет совместимость с де-факто существующими способами представления даты в большом количестве существующих редакторов электронных таблиц и имеет цель обеспечить совместимость и документированность сложившейся практики работы с электронными таблицами, не предлагая, в настоящий момент изменений, в силу неоднозначностей, которые могут возникнуть у миллионов пользователей в случае перехода на новую систему работы с датами.

62.	Часть 4, секция 3.17.7.341	-	te	<p>Описанная функция предписывает неправильное вычисление дней недели для некоторых дат 1900 года. Стандарт ISO не может предписывать использование некорректных значений для григорианского календаря. Это приведет к путанице, снижению возможностей для переносимости программ и неиссякаемому потоку ошибок.</p>	<p>См. разъяснения к замечаниям 57, 58.</p> <p>Ирония данного замечания состоит в том, что неверное предположение о том, что 1900 г. является високосным, было сделано в табличном процессоре Lotus 1-2-3, ныне принадлежащем IBM. Этот продукт был лидером рынка той эпохи, когда про Microsoft Excel еще никто не слышал. С тех пор все коммерческие табличные процессоры следуют этому неверному соглашению. Данное решение неоднократно обсуждалось ранее. Первоначальные разработчики, а за ними и многие другие, обычно оправдывали его тем, что это было полезно для повышения эффективности, и что действительно затронутым может оказаться намного менее, скажем, десяти тысячной доли процента документов (оценка приближительна). Позиция Microsoft изложена, в частности, в сообщении «Excel 2000 incorrectly assumes that the year 1900 is a leap year» (см. <a href="http://support.microsoft.com/kb/214326/en-us">http://support.microsoft.com/kb/214326/en-us</a>).</p> <p>Данное представление даты закрепляет совместимость с де-факто существующими способами представления даты в большом количестве существующих редакторов электронных таблиц и имеет цель обеспечить совместимость и документированность сложившейся практики работы с электронными таблицами, не предлагая, в настоящий момент изменений, в силу неоднозначностей, которые могут возникнуть у миллионов пользователей в случае перехода на новую систему работы с датами.</p> <p>Мнение о неиссякаемом потоке ошибок, кажется надуманным, т.к. данная система нумерации дат фактически используется подавляющим количеством пользователей электронных таблиц во всем мире и возможные ошибки уже стали частью стандартного поведения систем. Изменение данного поведения без отдельного глубокого переосмысления воздействия на огромное количество пользователей кажется необоснованным.</p> <p>Ведущие финансовые институты, участвовавшие в разработке ECMA-376 выразили сомнения в возможности изменения данного поведения в близком будущем.</p>
-----	----------------------------	---	----	--	---

63.	Часть 4, секция 3.2.29	-	te	<p>Описан алгоритм хеширования, предположительно основанный на алгоритме предыдущего поколения, использовавшемся в Excel. Этот алгоритм известен слабой устойчивостью, известны опубликованные способы его взлома.</p> <p>Безусловно, следует принять во внимание возражение, что в рамках OOXML устойчивость алгоритма хеширования не всегда существенна, так как пользователь может эффективно снять защиту, попросту распаковав zip-файл и отредактировав XML, чтобы удалить или заменить используемый ключ. Однако некоторые типы программ (например, интерфейс Google Documents), предоставляющие доступ на редактирование и/или чтение документов, способны в то же время обеспечить защиту документа, запретив физический доступ к его содержимому различными способами. С этой точки зрения доступ на редактирование документа не тождественен доступу к его XML-содержимому. Следовательно, имеет смысл применять для защиты документа устойчивый к взлому и в то же время переносимый алгоритм, например, SHA-256.</p>	<p>См. разъяснение к замечанию 15.</p> <p>Разработчики стандарта отдают себе отчет в росте угрозы взлома данного алгоритма специалистами, обладающими необходимыми навыками и средствами. В связи с этим они прямо предусмотрели в проекте следующее:</p> <ol style="list-style-type: none"> <li>1. Стандарт явным образом описывает использование следующих сильных алгоритмов (см. список в Части 4 стр.1166): MD2, MD4, MD5, SHA-1, MAC, RIPEMD, RIPEMD-160, HMAC, SHA-256, SHA-384, SHA-512.</li> <li>2. Стандарт описывает возможность расширения - подключение произвольного алгоритма хеширования (атрибут algIdExt (Cryptographic Algorithm Extensibility). Эта возможность необходима как для использования более сильных алгоритмов в связи с ростом вычислительных мощностей потенциальных взломщиков так и для соответствия нормативным требованиям в некоторых юрисдикциях.</li> <li>3. В комментариях разъясняется, как, вероятно, будет развиваться стандарт в дальнейшем в части использования более сложных, чем криптографические хеши, средств.</li> </ol>
-----	------------------------	---	----	--	---



64.	Часть 4, секция 3.2.29	стр. 1917-1922	te	<p>Не предоставлено нормативного описания алгоритма хеширования пароля, соответственно, невозможно сделать выводы о пригодности данной части стандарта для обеспечения совместимости между различными программными реализациями. Представленный в информативной секции код на языке C помечен как «пример» и включает архитектурно-зависимые битовые манипуляции.</p>	<p>См. разъяснения к замечаниям 15-17 касательно расширений и сильных криптохешей.</p> <p>Следует оговориться, что, как явно указано в документации, представленный в разделе 3.2.29 механизм установки парольной блокировки от изменений страницы (диапазона) не является средством обеспечения безопасности, а исключительно средством предотвращения непреднамеренных изменений.</p> <p>Приведенный пример реализации представляет интерес с точки зрения совместимости с существующими документами и будут использован преимущественно в этом контексте.</p> <p>Степень детализации может представляться авторам замечания недостаточной. Но она все же выше, чем в ODF, где описание алгоритма полностью отсутствует как в теле документа, так и в теле единственной применимой ссылки.</p>
65.	Часть 4, секция 3.2.29	стр. 1916	te	<p>Вероятным последствием реализации будет то, что при вводе пароля на некоторых национальных языках все символы будут интерпретированы как символ 0x3F, что делает парольную защиту бессмысленной и является неприемлемым.</p>	<p>Данное замечание забавным образом является просто цитатой из предупреждения относительно «простой», совместимой с унаследованными документами, версии крипто-хеша, содержащегося в спецификации OpenXML (см. Часть 4, секция 3.2.29 «workbookProtection (Workbook Protection)»): <i>«Code points with no representation in the target code page are replaced with Unicode character 0x3f»</i></p> <p>Такое поведение возможно для некоторых достаточно редких комбинаций языковых установок для некоторых символов в наборах Unicode (см. <a href="http://www.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/WindowsBestFit/readme.txt">http://www.unicode.org/Public/MAPPINGS/VENDORS/MICSFT/WindowsBestFit/readme.txt</a> ).</p> <p>Оно гарантировано невозможно для китайских, японских, корейских, вьетнамских, арабских, еврейских, кириллических, восточно- и западноевропейских, балтийских, турецких и ряда других кодировок.</p> <p>Да, это поведение не оптимально. Но именно таким оно было у старых документов. И именно их придется открыть и прочесть для конвертации.</p>

66.	Часть 4, секция 3.2.29	стр. 1916	te	<p>Описание алгоритма не определяет кодировку пароля. Предположительно, это Unicode, однако на практике это может быть Unicode с различным порядком следования байтов – UTF-6BE, UTF16-LE либо UTF16 с отметкой порядка байтов.</p> <p>Описанные алгоритмы используют манипуляции с данными на уровне байтов, реализация которых зависит от архитектуры аппаратного обеспечения и поддерживаемого ею порядка следования байтов. Необходимо, чтобы все предположения о порядке следования байтов были четко объявлены.</p>	<p>См. разъяснение к замечанию 16.</p> <p>На стр. 1916 разъяснено, что используется UTF-16. При этом алгоритм приведен в форме, не предполагающей ориентацию на определенный порядок байтов в машинном слове.</p>
67.	Часть 4, секция 3.3.1.61	-	te	<p>Набор допустимых значений атрибута pageSize охватывает не все варианты значений размера страницы, разрешенные стандартами ISO 216, ANSI 14Y.1, а также аналогичными JIS- и DIN-стандартами.</p>	<p>Действительно, в стандарте явным образом не перечислены все многочисленные возможные размеры бумаги, предусмотренные различными стандартами, а приведены наиболее часто употребляемые, в том числе, включающие размеры бумаги определенные в ISO 216.</p> <p>Однако проект стандарта позволяет задавать для листа произвольные размеры. Это делается путем использования атрибута usePrinterDefaults, что позволяет определить размеры бумаги допустимые текущим принтером, в том числе и любые нестандартные (не соответствующие вообще никакому стандарту) размеры, если принтер поддерживает такую возможность.</p> <p>Данное определение способа выбора размера бумаги не является ограничением, т.к. процесс редактирования электронных таблиц всегда происходит в предопределенном виде и не связан с их представлением на бумаге. Представление электронной таблицы на бумаге готовится приложением исходя из возможностей средств печати.</p>

68.	Часть 4, секция 3.3.1.69	-	te	<p>Не предоставлено нормативного описания алгоритма хеширования пароля, соответственно, невозможно сделать выводы о пригодности данной части стандарта для обеспечения совместимости между различными программными реализациями. Представленный в информативной секции код на языке C помечен как «пример» и включает архитектурно-зависимые битовые манипуляции.</p>	См. разъяснения к замечаниям 15, 16, 17, 63-66.
69.	Часть 4, секция 3.3.1.69	-	te	<p>Атрибут securityDescriptor, «определяющий учетные записи пользователей, имеющих право редактировать диапазон без предъявления пароля», является строковым значением. Спецификация не проясняет ни тип учетных записей, ни способ работы с ними, ни формат знака-разделителя. Неизвестно, подразумеваются ли в данном случае учетные записи пользователей на локальной машине или, например, в каталоге LDAP. Неизвестно, разделяются ли имена учетных записей запятой, точкой с запятой либо каким-то другим символом. Такое описание исключает совместимость программных реализаций.</p>	<p>В задачу разработчиков ECMA-376 не входит амбициозная цель создания универсального стандарта цифровой идентичности (digital identity). Это – важная и сложная в концептуальном, технологическом и даже, отчасти, в политическом, аспектах задача, которую безусловно еще предстоит решить (определенные попытки уже были сделаны).</p> <p>Предполагается, что разработчики приложений выберут те механизмы, которые реально имеются в их распоряжении на целевых платформах.</p> <p>Следует также отметить, что данная опция характерна не столько для документов долговременного хранения, сколько для документов, создаваемых в ходе оперативных бизнес-процессов (возможно даже генерируемых бизнес-приложениями).</p> <p>Представляется, что ODF также не предлагает определенного способа решения данной проблемы – даже просто способа сохранить соответствующее значение.</p>

70.	Часть 4, секция 3.3.1.69	-	te	<p>Описан алгоритм хеширования, предположительно основанный на алгоритме предыдущего поколения, использовавшемся в Excel. Этот алгоритм известен слабой устойчивостью, известны опубликованные способы его взлома.</p> <p>Безусловно, следует принять во внимание возражение, что в рамках OOXML устойчивость алгоритма хеширования не всегда существенна, так как пользователь может эффективно снять защиту, попросту распаковав zip-файл и отредактировав XML, чтобы удалить или заменить используемый ключ. Однако некоторые типы программ (например, интерфейс Google Documents), предоставляющие доступ на редактирование и/или чтение документов, способны в то же время обеспечить защиту документа, запретив физический доступ к его содержимому различными способами. С этой точки зрения доступ на редактирование документа не тождественен доступу к его XML-содержимому. Следовательно, имеет смысл применять для защиты документа устойчивый к взлому и в то же время переносимый алгоритм, например, SHA-256.</p>	См. разъяснения к замечаниям 15, 16, 17, 63-66.
71.	Часть 4, секция 5.1.12.28	-	te	<p>Этот тип используется лишь в двух местах — в разделах 5.1.2.2.32 и 5.1.2.2.33, в обоих случаях для представления цветового кода RGB. В то же время в стандарте определен тип ST_HexColorRGB, очевидно подходящий для данной ситуации.</p>	<p>ST_HexColorRGB - простой тип определенный в пространстве имен WordprocessingML, в то время как ST_HexBinary3 - простой тип, определенный в пространстве имен DrawingML. Т.е. эти два типа определены в различных пространствах имен и поэтому ST_hexBinary3 не может быть исключен из описания.</p>

72.	Часть 4, секция 5.1.12.37	-	te	Отсутствует определение «Классификации Panose» («Panose classification») для шрифта.	<p>См. разъяснение к возражениям 7, 51.</p> <p>Данный комментарий, возможно, основан на неполном прочтении необходимых разделов спецификации OOXML. Как указано в Части 4 при определении соответствующих элементов (panose1) в проекте стандарта используется классификация Panose-1 и дана явная отсылка к спецификации PANOSE Classification Guide, Version 1.2.</p> <p>Интересно отметить, что в ODF 1.0 panose явно упоминается только в использовании атрибутов (стр. 478, <i>&lt;attribute name="svg:panose-1"/&gt;</i>) и никак не определяется. Возможно, необходимо использовать косвенные ссылки через спецификацию SVG.</p>
73.	Часть 4, секция 5.1.12.37	-	te	Предписывается использовать значение Panose, чтобы позволить приложению при необходимости определять наиболее близкий к запрошенному шрифт. В то же время не описывается метрика соответствий шрифтов либо эвристический алгоритм определения соответствия, которые позволили бы реализовать эту функциональность.	См. разъяснение к возражениям 7, 51, 72.

74.	Часть 4, секция 5.1.3.4	-	te	<p>Описывается присоединение видео в формате QuickTime к объекту презентации. Не предоставлено определение формата QuickTime. Без спецификации формата и поддерживаемых библиотек декомпрессии (кодеков) совместимость программных реализаций стандарта невозможна.</p>	<p>QuickTime является широко распространенным мультимедийным форматом, предложенным Apple (<a href="http://www.apple.com/quicktime/">http://www.apple.com/quicktime/</a> и завоевавшим исключительно широкую популярность ). Его определение, очевидно, находится вне рамок спецификации OOXML. В то же время его ВОЗМОЖНОЕ ИСПОЛЬЗОВАНИЕ детально определено в спецификации.</p> <p>Open XML не стремится ограничивать возможности приложений, определяя их поведение на основе стандарта. Секция 2.6 - "Interoperability Guidelines" гласит: «... приложение должно иметь сопроводительную документацию, которая описывает поддерживаемое подмножество Стандарта. Документация должна явно описывать поведение приложения, которое в случае отсутствия такой документации, выглядело бы как нарушающее семантику Стандарта»</p> <p>Следуя логике авторов замечания можно придти к выводу о серьезном ограничении интероперабельности и функциональных возможностей ISO 26300 (ODF) – там даже не упоминается QuickTime. Следовательно он не сможет и использоваться?</p>
-----	-------------------------	---	----	---	--

75.	Часть 4, секция 6 -	-	te	<p>Определяется формат Vector Markup Language (VML), который, в дополнение к DrawingML, предоставляет свой словарь описания графических объектов. В секции 6.1 утверждается, что «DrawingML – более новый формат с более широкими возможностями, созданный с целью полной замены VML в документах Office Open XML. VML должен рассматриваться как устаревший формат, включенный в данный стандарт только в целях обеспечения совместимости с форматами предыдущего поколения. При необходимости использовать в новых приложениях язык, описывающий графические объекты, настойчиво рекомендуется использовать DrawingML». Необходимость поддержки VML в реализациях стандарта приведет к значительному росту затрат на реализацию (спецификация VML занимает более 600 страниц), поставит независимых производителей ПО в проигрышное положение по сравнению с Microsoft и дополнительно затруднит разработку совместимых программных реализаций стандарта.</p>	<p>См. разъяснение к замечанию 4.</p> <p>В замечании дается вольная (расширительная) трактовка положений проекта стандарта, в частности секции 8.6.2 Части 1. DrawingML является, безусловно, более «богатым» и более «прогрессивным» форматом векторной графики и рекомендуется проектом стандарта к применению для описания векторных графических изображений. Однако для простых задач, не связанных непосредственно с созданием сложных изобразительных объектов, средства VML обладают оптимальной выразительностью и могут быть легко реализованы. Это также значительно облегчает достижение обратной совместимости.</p> <p>Поддержка предыдущих версий документов (принцип обратной совместимости) является фундаментальным принципом стандарта Open XML. VML важен для представления векторной графики в старых версиях документов и, таким образом, важен для стандарта Open XML. Тем не менее, для генерации новых документов рекомендуется избегать использования VML, что и объясняет его обозначение в спецификации, как «нерекомендуемого к использованию» (deprecated). Документирование данной функциональности, как нерекомендованной к использованию, также важно для целей долгосрочного архивирования документов предыдущих версий. Обозначение функций, как нерекомендованных к использованию (в т.ч. в будущем), является общепринятой практикой стандартизирующих организаций, применяемой для подобных целей.</p>
-----	---------------------	---	----	---	--

76.	Часть 4, секция 6	стр 4343-4960	te	<p>Все подразделы секции 6 описывают материал, определенный разработчиками стандарта как «устаревший», что делает всю секцию устаревшей. Представляется нецелесообразным включать в стандарт ISO устаревшие разделы.</p>	<p>Замечание является следствием фундаментального непонимания его авторами процесса и целей стандартизации. Как говорится в ISO/IEC Guide 2:2004: «<i>Standards should be based on the consolidated results of science, technology and experience, and aimed at the promotion of optimum community benefits</i>».</p> <p>Действительно широкоупотребимые и полезные стандарты, как правило, имеют историю, отражают предыдущий опыт, и определяют оптимальные направления будущего развития. Так проект стандарта OpenXML имеет разделы:</p> <ul style="list-style-type: none"> <li>• определяющие ключевые современные решения, рекомендуемые или обязательные к реализации;</li> <li>• описывающие используемые спецификации, которые, возможно, в будущем переведены в разряд устаревших. Тем не менее их поддержка может быть достаточно длительной, либо практически постоянной из-за длительного жизненного цикла некоторых категорий документов;</li> <li>• рассматривающих наиболее вероятные направления развития и расширения.</li> </ul> <p>Последовательное применение такого подхода при разработке проекта стандарта 29500 является исключительно важным для целей достижения реальной интероперабельности и открытости, подтверждением различных областей применения OpenXML и ODF и еще одним аргументом в пользу принятия стандарта ISO/IEC 29500. Поддержка предыдущих версий документов (принцип обратной совместимости) является фундаментальным принципом стандарта Open XML. VML важен для представления векторной графики в старых версиях документов и, таким образом, важен для стандарта Open XML. Тем не менее, для генерации новых документов рекомендуется избегать использования VML, что и объясняет его обозначение в спецификации, как «нерекомендуемого к использованию» (deprecated). Документирование данной функциональности, как нерекомендованной к использованию, также важно для целей долгосрочного архивирования документов предыдущих версий. Обозначение функций, как нерекомендованных к использованию, является общепринятой практикой стандартизирующих организаций, применяемой для подобных целей.</p>
-----	-------------------	---------------	----	--	--



77.	Часть 4, секция 6.1.2.19	стр. 4653, «equationxml»	te	<p>Описывается атрибут «equationxml» для элементов «shape», «используемый для восстановления уравнений с использованием синтаксиса Office Open XML Math». Поскольку «фактический формат содержимого атрибута определяется приложением», уверенный обмен данными между различными приложениями становится невозможным. Если даже проигнорировать существование MathML и поставить целью создание нового диалекта XML для описания математических формул, имело бы смысл как минимум определять его как полноценное подмножество XML (а не значение атрибута) и избегать обработки данных по зависимым от приложения алгоритмам.</p>	<p>Говоря кратко, речь идет о возможности сохранить в выходном документе (в формате, возможно, отличном от OpenXML) формулы в виде, отличном от Open XML Math (например в виде изображений или даже MathML). При этом данный тег позволяет указать, какую информацию можно сохранить в качестве «альтернативного описания формулы», чтобы не потерять всю ее необратимо при преобразовании. Это может быть разметка Office Open XML Math или любая другая.</p> <p>Данная функциональность обеспечивает альтернативное представление зоны математического выражения. Формат разметки для математических символов полностью определен в части 4 секция 7.1. Если приложение, использующее документ, не поддерживает данную часть стандарта, то генерирующее приложение (формирующее документ в формате Open XML) может предоставить изображение математического выражения, которое позволит приложению, использующему документ, корректно отобразить это выражение. Данный атрибут позволяет сохранить математические символы в формате, определенном в секции 7.1, или в любом другом формате, так что при сохранении документа с изображением математического выражения, информация не теряется.</p> <p>Данная опция является позитивным примером заботы разработчиков проекта о реальной интероперабельности.</p>
-----	--------------------------	--------------------------	----	--	--

78.	Часть 4, секция 6.1.2.19	стр. 4655, «gfxdata»	te	<p>Описывается атрибут «gfxdata» для элементов «shape», «представляющий собой закодированное по алгоритму base-64 содержимое в формате DrawingML». Поскольку «содержимое пакета определяется приложением», даже при условии «использования частей данного стандарта там, где это представляется возможным», этой информации недостаточно для того, чтобы данные могли быть корректно обработаны, а содержимое DrawingML могло быть корректно отображено любой программой, представляющей независимую реализацию стандарта. Если даже проигнорировать существование SVG и поставить целью создание полностью нового диалекта XML для описания графической разметки, имело бы смысл как минимум определять его как полноценное подмножество XML (а не значение атрибута) и избегать обработки данных по зависимым от приложения алгоритмам.</p>	<p>Авторы замечания, возможно, не обратили внимания на то, что DrawingML подробно описан в Части 4 в Секции 5 «DrawingML Reference Material», для него введено пространство имен, и, таким образом, их пожелание «...имело бы смысл как минимум определять его как полноценное подмножество XML (а не значение атрибута)» уже выполнено.</p> <p>Употребление тега gfxdata ограничено, а смысл его состоит исключительно в том, чтобы позволить приложению, использующему VML, и не желающему обрабатывать DrawingML, тем не менее сохранить корректную информацию в этом формате и, возможно, предупредить третье приложение о ее наличии.</p> <p>Также в стандарте не определен алгоритм обращения с блоком закодированных данных для приложения, не поддерживающего DrawingML (возможно, оно не будет его обрабатывать, или вызовет внешний обработчик). И это совершенно логично. А вот интерпретация DrawingML приложением, декларировавшим его поддержку в стандарте, конечно, описана.</p> <p>По мнению одного из авторов разъяснений, определение в проекте стандарта 29500 объектов и пространства имен DrawingML имеет определенные преимущества перед решением, принятым в ODF (ISO 26300), где SVG предлагается к реализации с добавлениями и изъятиями. Потенциально это означает, что в некоторых случаях приложения, разработанные для поддержки OpenXML, будет даже легче интегрировать с продуктами (библиотеками, фильтрами) SVG, чем приложения, изначально разработанные для поддержки ODF с SVG.</p> <p>Следует также повторить, что Open XML не стремится ограничивать возможности приложений, определяя их поведение на основе стандарта. Секция 2.6 “Interoperability Guidelines” гласит: «... приложение должно иметь сопроводительную документацию, которая описывает поддерживаемое подмножество Стандарта. Документация должна явно описывать поведение приложения, которое в случае отсутствия такой документации, выглядело бы как нарушающее семантику Стандарта».</p>
-----	--------------------------	----------------------	----	---	--

79.	Часть 4, секция 6.2.2.14	-	te	<p>Описывается элемент «ink» («чернильная метка»), хранящий «чернильные пометки в формате, определяемом приложением». По всей видимости, имеются в виду сделанные с помощью устройств ввода типа «tablet» рукописные комментарии к электронным документам. Эти аннотации зачастую являются важной частью документа, и их формат не определен в стандарте OOXML. Представляется неприемлемым включение в стандарт элементов, предназначенных для хранения данных в закрытых частных форматах без одновременного описания формата, позволяющего хранить эти данные таким образом, чтобы обеспечить их переносимость между независимыми реализациями.</p> <p>Определить формат «ink» либо исключить описание элемента из стандарта, реализовывая его в качестве расширения.</p>	<p>Определение формата хранения данных рукописного ввода понимается выходит за рамки задач данного стандарта. В то же время полностью умолчать о наличии такой возможности в стандарте представляется неоптимальным решением. Элемент ink включен в стандарт, чтобы предоставить интероперабельный способ предупредить третьи приложения о наличии данных рукописного ввода.</p> <p>Предложение реализовать данный элемент как «кастомное» расширение, а также аналогичные предложения по поводу других элементов поддержки интероперабельности, нельзя признать целесообразным. Оно приведет к быстрому и трудно управляемому росту расширений, исключительная цель которых, - предупреждения о совместимости. Если такой подход еще приемлем для относительно редко встречающихся опций, то он быстро создаст серьезные проблемы для распространенных приложений/возможностей и потребует. Потребуется дополнительный стандарт, дополнительный регулирующий орган, какое-то иное решение? В частности, авторы разъяснений опасаются, что эта проблема затруднит реальное использование ODF.</p> <p>Как уже было отмечено в разъяснении к замечаниям 78 и др., с целью расширения возможностей использования, спецификация Open XML декларирует элемент или атрибут и оставляет его реализацию приложению. Секция 2.6 «Interoperability Guidelines» гласит, что «Если приложение перемещает, добавляет, модифицирует или удаляет экземпляры элементов, влияя на семантику документа, то приложение обязано декларировать данное поведение в своей документации.» Далее секция 2.6 утверждает, что если элемент или атрибут является определяемым приложением, то он должен сопровождаться документацией, описывающей его назначение и функциональность.</p>
-----	--------------------------	---	----	--	--

80.	Часть 4, секция 6.4.2.10	-	te	<p>Элемент определен как «элемент общего использования для объектов, которые используют представление изображения, таких как OLE, встроенные элементы управления, камеры, строки подписи». Тем не менее, отсутствуют ссылки на описания форматов, которым соответствуют разрешенные для элемента значения (EMF, WMF и др.)</p>	<p>Говоря коротко речь идет об опциональной возможности для приложения показать некоторое изображение вместо ole-объекта, если оно его «не поняло» или «не хочет» поддерживать соответствующую функциональность внедрения.</p> <p>В данной секции определяется рекомендуемое поведение приложения, но не требование соответствия Стандарту. Open XML не предъявляет требований к использованию WMF или EMF. Секция 6.2.3.17 позволяет использовать любой графический формат. Допустимые значения полей – "Bitmap", "Pict", "PictOld", "PictPrint", and "PictScreen" могут отображаться на любой целевой графический формат. WMF и EMF приведены в качестве возможных вариантов.</p> <p>Следует заметить, что здесь как OpenXML так и ODF следуют одинаковой логике. Так в п. 9.3.3. спецификации ODF 1.0 дается следующее определение тега draw:ole-object: «It is recommended to include an image representation of the object into the frame in addition to the object itself».</p>
81.	Часть 4, секция 6.4.2.10	-	te	<p>Перечисленные форматы специфичны для платформы Windows. Они не поддерживаются в других операционных системах. В частности, в Linux копирование изображения в буфер обмена производится с использованием открытого стандартного формата, например, PNG.</p>	<p>В данной секции (в контексте работы с объектами, встраиваемыми в электронные таблицы) речь идет исключительно о рендеринге представления объектов в клипборде, если они имеют альтернативное визуальное представление (типа OLE, элементов управления, камер, строк подписи и пр.). Указанная секция определяет рекомендуемое поведение приложения, но не требование к соответствию Стандарту.</p> <p>Open XML не предъявляет требований к использованию конкретного графического формата. Список перечисляет возможные варианты, включая bitmap, изображение для представления на экране, изображение для представления на печати, WMF, EMF. Это поведение не накладывает ограничения на формат файлов изображений и полностью расширяемо. В частности, можно использовать PNG.</p>

82.	Часть 4, секция 7.1	-	te	<p>Представлена спецификация языка разметки Office Open Math Markup Language, подмножества XML, предназначенного для описания математических уравнений. Этот диалект решает ту же задачу, что и стандарт W3C MathML, с одной стороны обладающий большей историей, с другой стороны – активно развивающийся в настоящее время. С учетом того, что механизм обработки уравнений в программах семейства Microsoft Word был полностью переработан для Word 2007, аргумент о необходимости введения собственного подмножества XML с целью обеспечения совместимости с форматами Word предыдущих поколений не выглядит убедительным.</p>	<p>MathML не является стандартом ISO и от стандарта Ecma-376 не требуется хранить математические выражения, используя данную рекомендацию. При том, что многие возможности этих спецификаций схожи, их базовые концепции различаются.</p> <p>Подробное сравнение MathML и OMML не может быть проведено в рамках формата «Разъяснений к замечаниям». Отметим только, что MathML использует инфиксную нотацию, а OMML – позиционную. Кроме того, OMML позволяет пользователю внедрить произвольный текст в зону математического выражения, а MathML менее гибок в этом отношении. Как следствие, реализующее OpenXML с OMML приложение сможет естественным образом обращаться с MathML. Ну и традиционный аргумент требования обратной совместимости с огромным массивом унаследованных документов здесь тоже применим.</p> <p>Стоит отметить, что так как и OMML и MathML являются XML спецификациями, то существуют XSLT шаблоны для преобразования одного формата в другой.</p>
-----	---------------------	---	----	--	--

83.	Часть 4, секция 7.4.2.4	-	te	<p>Наличие символов, не предусмотренных спецификацией XML (вне зависимости от того, экранируются ли они с точки зрения синтаксиса OOXML), нарушает совместимость со стандартом XML и инструментами, основанными на нем. Группа W3C по интернационализации (W3C Internationalization Activity) подтверждает такую трактовку, утверждая, что «контрольные символы должны быть заменены соответствующей разметкой. Поскольку XML предоставляет стандартный способ кодирования структурированных данных, использование каких-либо контрольных кодов, помимо разметки, нивелирует преимущества использования XML. Использование контрольных кодов в HTML и XHTML не может служить аргументом, так как эти языки предназначены для представления исключительно текста, а не произвольных данных».</p>	<p>Стандарт избегает использования не-XML символов. Группа W3C по интернационализации (W3C Internationalization Activity) также утверждает, что «Если данные являются не текстовыми, а двоичными, то более практично закодировать их, например, с использованием base64 или как шестнадцатеричные значения, для гарантии использования только поддерживаемого набора символов в тексте разметки».</p> <p>Тип bstr может применяться в пользовательских (custom) свойствах, для которых явно предполагается пользовательское определение типа. Тип bstr кодируется в Unicode формате _хНННН_, где Н представляет шестнадцатеричный код символа.</p> <p>Таким образом противоречия с рекомендациями W3C и духом XML нет.</p>
-----	-------------------------	---	----	---	--

84.	Часть 4, секция 7.4.2.4	-	te	<p>Определяется новый тип XML-строк, который позволяет включения экранированных символов Unicode, что недопустимо с точки зрения синтаксиса XML. Кроме того, любая спецификация механизма экранирования символов должна включать в себя способ экранирования экранирующего символа. Буквальный пример, приведенный в 7.4.2.4, этого не демонстрирует.</p>	<p>Стандарт избегает использования не-XML символов. Группа W3C по интернационализации (W3C Internationalization Activity) также утверждает, что «Если данные являются не текстовыми, а двоичными, то более практично закодировать их, например, с использованием base64 или как шестнадцатеричные значения, для гарантии использования только поддерживаемого набора символов в тексте разметки».</p> <p>Буквально это и требуется описанием элемента bstr. Он может применяться в пользовательских (custom) свойствах, для которых явно предполагается пользовательское определение типа. Экранированные символы для типа bstr кодируются в Unicode формате <code>_xHNNH_</code>, где H представляет шестнадцатеричный код символа, что соответствует правилам XML.</p> <p>Можно также отметить, что допустимые значения элемента bstr определены в типе данных строки соответствующей XML схемы.</p> <p>Если национальный орган стандартизации сочтет целесообразным, то он может переqualифицировать замечание в «re» (редакционное) и представить (или рекомендовать представить) дополнительный (или другой) пример.</p>
-----	-------------------------	---	----	---	---

85. при мер.	Часть 4, секция 7.4.2.5	-	te	<p>Вызывает существенные сомнения представление строк как null-terminated строк языка C, закодированных по алгоритму base-64. Это расходится с синтаксисом XML и существенно снижает совместимость с программными инструментами для обработки XML.</p>	<p>В замечании речь идет об очень специальной ситуации – способе получить (возможно от внешнего приложения) информацию для регистрации имени произвольного не известного заранее типа формата данных в клипборде (из чего, кстати, видно, что формат действительно может быть произвольным). Значение атрибута format элемента cf и находящиеся в буфере данные не имеют влияния на то, как отображается документ, и могут быть «интересны» использующему их приложению на момент времени исполнения (run time).</p> <p>В стандарте описан способ сравнения регистрируемых имен (нечувствительность к регистру, выбор кодовой базы и пр.), подобранный так, чтобы сделать алгоритм максимально широко применимым на разных платформах и разными приложениями. Это позволит разным приложениям использовать один набор данных буфера, «понимая» в каком формате помещены в него данные.</p> <p>Само по себе использование для целей передачи имени, строки, закодированной по base64, представляется вполне естественным. Разумеется, ODF также использует base 64 при кодировке бинарных строк. Соглашение об использовании C-style null-terminated строк совершенно естественно по крайней мере для Unix, Linux и Windows систем, соответствует стандартам C, C++ и Posix и никак не сказывается на возможности валидации или иной обработки XML файлов.</p>
--------------------	----------------------------	---	----	--	---



86.	Часть 4, секция 7.4.2.5	-	te	<p>Значение -3 определяет GUID, содержащий идентификатор формата (FMTID). Форматы ни для GUID, ни для FMTID не определены.</p>	<p>GUID (FMTID) – это 128-битное целое число (16 байт), имеющее крайне низкую вероятность повторяемости. Это значение определяется форматом и каждый формат имеет свой FMTID. Проект Стандарта не налагает ограничений на то, как приложения генерируют GUID, только на его размеры (и в этом смысле формат GUID определен в 7.4.2.5 и других местах как 32 16-ричных числа) и способы использования. Следовательно, описание формата GUID находится вне рамок Стандарта.</p> <p>[Примечание: GUID является реализацией UUID, определенного OSF как часть DCE. См. в частности - <a href="http://www.itu.int/ITU-T/studygroups/com17/oid.htm">http://www.itu.int/ITU-T/studygroups/com17/oid.htm</a> Если приложение или пользователь не генерируют GUID, они могут использовать внешний сервис. См., например, <a href="http://www.itu.int/ITU-T/asn1/uuid.html">http://www.itu.int/ITU-T/asn1/uuid.html</a> . Однако все это находится, очевидно, за рамками данного Стандарта]</p>
87.	Часть 4, секция 7.4.2.5	-	te	<p>Элемент определяет значения, которые могут быть использованы только на платформах Windows или Macintosh, но не могут быть использованы в Linux или любой другой операционной системе.</p>	<p>Секция 7.4.2.5 определяет формат данных буфера обмена. Атрибут format элемента cf может принимать значения -3, -2, -1, 0 или любое целое положительное число. Положительное целое число может быть использовано для регистрации определенного формата в простом типе ST_CF (см. обсужден е выше в разъяснении к замечанию 85).</p> <p>К сожалению, авторы замечания не поняли суть критикуемого ими пункта. Очевидно, в буфер могут быть помещены данные в любом формате для любой операционной системы. Секция 7.4.2.5 вообще не налагает на это никаких ограничений! Проблема состоит в том, как сообщить информацию об используемом заранее неизвестном формате неизвестного приложения на неизвестной платформе некоторому третьему приложению, чтобы оно смогло использовать тот же буфер. В отличие от ODF, проект стандарта OpenXML предлагает решение, основанное на кодировании имени формата в base-64 нуль-терминированных C-строках и «мягком» сравнении, которое можно выполнить на практически любой платформе, включая все ОС, совместимые со стандартами IEEE 1003 : ISO/IEC 9945 Posix (AIX, HP-UIX, QNX, Solaris, Windows и др.) и большинство несовместимых или «не совсем совместимых» (Linux, FreeBSD, Z/OS и многие другие).</p>

88.	Часть 4, секция 7.4.2.5	-	te	<p>Даже в рамках одной операционной системы информации недостаточно, чтобы обеспечить совместимость различных реализаций. Например, не определено, какие допустимые значения и какую смысловую нагрузку может иметь «значение формата встроенного буфера обмена Windows» («built-in Windows clipboard format value»).</p>	<p>См. также разъяснение к замечанию 87.</p> <p>Комментарий, возможно, основан на неверной интерпретации спецификации. Секция 7.4.2.5 определяет формат данных буфера обмена и значения должны быть -3, -2, -1, 0 или любое целое положительное число. В случае использования встроенного формата буфера обмена Windows значение атрибута format элемента cf будет -1 !</p> <p>Если же авторов замечания интересует формат данных хранимых в буфере обмена на платформе Windows, то они должны искать ее соответствующей документации по платформе. Буфер обмена Windows широко используется тысячами приложений разных поставщиков и с ним не связано никаких проблем. Аналогично на других платформах, как правило, также можно найти необходимую документацию.</p> <p>Здесь, возможно, следует повторить, что значение атрибута format элемента cf и находящиеся в буфере данные не имеют влияния на то, как отображается документ, и могут быть «интересны» использующему их приложению на момент времени исполнения (run time).</p>
-----	-------------------------	---	----	---	--

89.	Заключение	-	te	<p>Название «Office Open XML» зачастую ошибочно пишется как «Open Office XML», создавая ложное впечатление связи с проектом Open Office. Данная путаница в наименованиях имеет документальные подтверждения в ряде случаев, включая аналитические исследования, а также собственные пресс-релизы и блоги корпорации Microsoft. Учитывая шестилетний приоритет проекта OpenOffice, ECMA следует выбрать другое имя для своего стандарта, которое не вызвало бы путаницы.</p>	<p>Замечание является тенденциозным и, к сожалению, не совсем компетентным.</p> <p>Если где либо в тексте ECMA-376 встречается словосочетание «Open Office XML» (не «open Office XML!»), следует указать на это, как на редакционную («ге») ошибку и предложить исправление в установленном ISO JTC-1 порядке. Если же речь идет о подобных ошибках за пределами данной спецификации, то полезно на них указывать, но эта тема находится за рамками настоящего документа.</p> <p>Трудно с полной определенностью сказать, какой смысл авторы замечания вкладывают в термин «проект Open Office». Вероятно, речь идет о проекте разработки продуктов OpenOffice.org (см. <a href="http://www.openoffice.org">www.openoffice.org</a> ).</p> <p>Термин «Office» не является торговой маркой или защищенным товарным знаком, сочетание слов «Open Office» не является зарегистрированным товарным знаком т.н. «проекта Open Office». Более того, известно, что «Open Office», вероятно, является зарегистрированным товарным знаком, не имеющим отношения к т.н. «проекту Open Office» (см. <a href="http://marketing.openoffice.org/servlets/ReadMsg?list=dev&amp;msgNo=20640">http://marketing.openoffice.org/servlets/ReadMsg?list=dev&amp;msgNo=20640</a> ). Вообще, следует отметить, что разработчики программного продукта OpenOffice.org никогда не называют его «Open Office» и предостерегают от этого других.</p> <p>Наименование «Ecma Office Open XML File formats» является однозначным и недвусмысленным, так же как и «Office Open XML». Встречающиеся сокращения и альтернативные наименования могут быть однозначными и наглядными, но не имеют отношения к Стандарту, который использует наименования, не вызывающие никакой путаницы.</p>
-----	------------	---	----	---	---

90.	Заключение			<p>Из общего контекста документа очевидно, что разработчики не сделали попытки развить стандарт OOXML на базе ранее принятого ISO стандарта представления документов в XML под названием ODF ISO/EIC 26300:2006. Поскольку разделение стандартов и увеличение количества необоснованных расхождений в них выглядят весьма нежелательными, представляется логичным переписать спецификацию OOXML как расширение существующего стандарта.</p>	<p>В период первой части процесса рассмотрения в ISO проекта стандарта ECMA-376, т.е. так называемого Contradiction Period, ISO/IEC JTC-1 принял решение, что не существует проблем в отношении «расхождений» между DIS 29500 и другими стандартами, что могло бы потребовать изменений в поданном на рассмотрении DIS 29500. Следовательно, не было необходимости изменений DIS 29500 перед началом 5-месячного процесса голосования.</p> <p>По мнению авторов разъяснений к замечаниям OpenXML и ODF имеют существенно разные области применения и построены на принципиально различных архитектурных принципах. При этом OpenXML отличается глубокой проработкой деталей, полнотой охвата форматов и сценариев обработки документов, пониманием реальных проблем интероперабельности. Даже до стандартизации в ISO/IEC это – действительно зрелый и полезный стандарт. Жаль, что некоторые его передовые черты не нашли отражения в ODF.</p> <p>В любом случае реальное будущее за поддержкой приложениями многих форматов и даже возможностью сохранения документов в мультиформатном режиме.</p> <p>Именно поэтому важно, чтобы все основные офисные форматы – OpenXML, ODF, PDF и UOF (формат, разработанный при поддержке правительства КНР) были своевременно поставлены под контроль международного сообщества.</p>
-----	------------	--	--	---	--