

## СРЕДСТВА ХРАНЕНИЯ И АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ ВЕРСИЙ ТЕКСТОВОЙ ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ

С.А. Леоновец  
Университет ИТМО  
ser2694@yandex.ru

### Введение

Процесс разработки бортового приборного оборудования для авиации сопровождается подготовкой конструкторской и программной документации на различных этапах проектирования: эскизное проектирование; технический проект; технические предложения; рабочее конструкторское проектирование. Перечень разрабатываемых документов регламентируется соответствующими государственными стандартами, например, ГОСТ 2.102-68, и определяется сложностью системы и её практической области применения. Данная техническая документация имеет свой жизненный цикл, на протяжении которого необходимо следить за статусом инженерных данных, использованных в ней. В распоряжении разработчиков проектных организаций сегодня имеются различные программные средства, помогающие осуществить контроль версий документов, например, Git. Но они не учитывают специфику действующих государственных стандартов.

Поэтому целью работы являлось написание программного обеспечения, которое предоставило бы пользователям возможность автоматизированной подготовки технической документации, а также осуществлять контроль версий.

### Система контроля версий

Для решения данной задачи была среда Microsoft Visual Studio. Программное написано на языке C# с использованием WPF. В состав САПР входит система контроля версий, которая осуществляет постоянный контроль данных документа в ходе всего жизненного цикла [2].

При создании документа автоматически переходит в состояние «черновик» (draft) с версией 1.0. Это состояние используется для редактирования. По окончании редактирования версия документа подвергается формальной инспекции, для чего переводится в состояние «проверка» (proposed). С этого момента редактирование запрещается. По результатам проверки версия документа либо становится новой базовой версией единицы конфигурации (ЕК), переходя в состояние утверждено (approved), либо отправляется на доработку, переходя в состояние «отклонен» (declined). В обоих случаях версия документа фиксируется для истории и ее редактирование не допускается. При необходимости внести изменения выполняется

порождение следующей версии документа. При этом в зависимости от состояния последней версии документа (approved или declined) присвоение номера версии для порождаемой версии документа.

Все версии документа сохраняются. Обеспечена возможность просмотра любой, выбранной пользователем, версии документа, а также возможность сравнения двух любых версий одного документа.

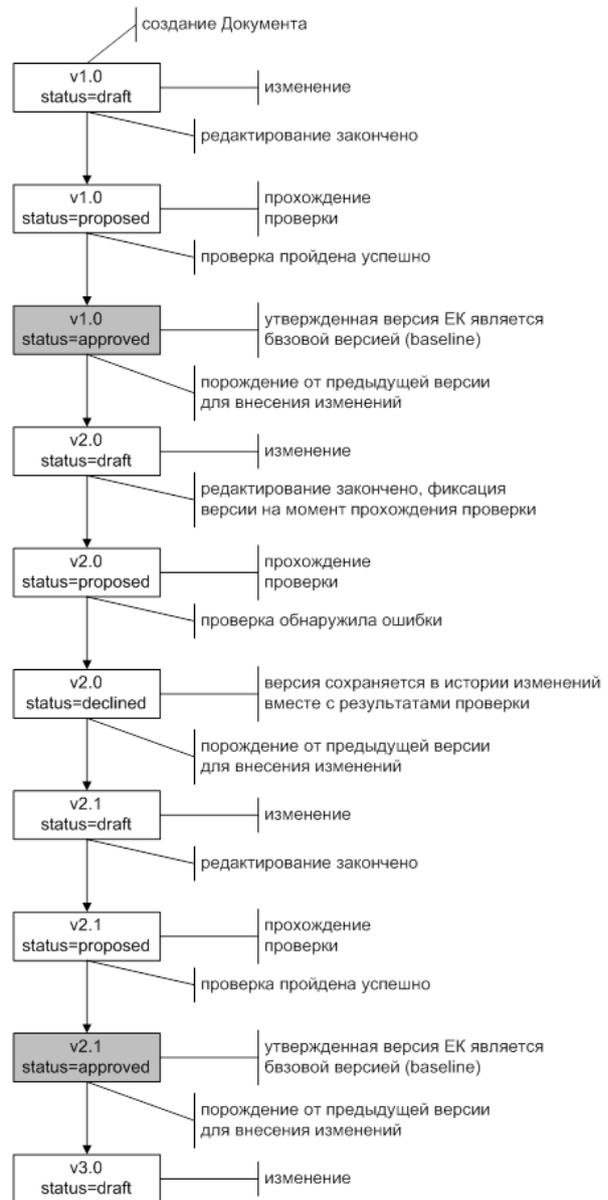


Рис. 1. Пример «эволюции» документа

Интерфейс разработанной системы контроля версий имеет древовидную наглядную структуру и представлен на рис.2.

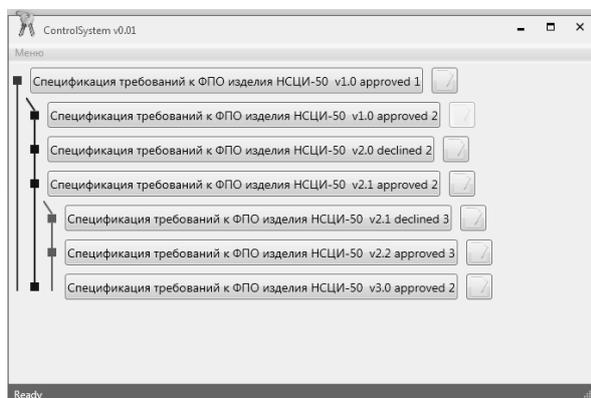


Рис. 2. Интерфейс системы контроля версий

Было проведено исследование скорости сохранения документов в разных форматах с использованием библиотеки *Microsoft.Office.Interop.Word.dll* версии 14.0, а также языка C#. Тестирование скорости сохранения производилось замером времени выполнения следующего кода:

```
Microsoft.Office.Interop.Word._Application
appWord = new
Microsoft.Office.Interop.Word.Application();
// Заполнение документа контентом
CreateAndFillActiveDoc(appWord);
object fileName;
object FileFormat =
Word.WdSaveFormat.wdFormatRTF;
appWord.ActiveDocument.SaveAs(ref
fileName, ref FileFormat);
appWord.Quit();
```

Расширение сохраняемого документа задается переменной *FileFormat*. В приведенном выше примере выбран формат *RTF*. Также были исследованы форматы: *PDF, XPS, DOC, DOCX, ODT, MHTML*. Сохранялись данные объемом 3 страницы A4, содержащие таблицу, изображение и текст. Результаты эксперимента представлены в таблице.

Таблица. Скорость сохранения данных при различных форматах документов

Формат документа	Время сохранения, мс
DOC	161
DOCX	155
XPS	328
PDF	356
RTF	60
ODT	101
MHTML	115

Исходя из полученных результатов, был сделан вывод, что сохранять версии каждого документа целесообразно в формате *RTF*, это позволит сэкономить время пользователя при сохранении большого объема данных.

### Заключение

Предложен новый компонент системы автоматической генерации конструкторских и программных документов – подсистема контроля версий, обеспечивающая непрерывность контроля жизненного цикла инженерных данных, с целью повышения качества документов. Пользователь в любой момент времени может проследить как менялся документ, кто вносил определенные изменения и когда, а также «откатиться» к прошлой версии.

Полученные результаты могут быть использованы разработчиками проектных организаций, которые оформляют техническую документацию по ЕСКД и ЕСПД. Апробация работы данной системы может производиться с использованием действующих в области авиационного приборостроения отечественных стандартов и стандартов *ARINC* (*Aeronautical Radio Inc., США*). Автоматизация процесса подготовки конструкторской документации существенно сокращает трудоемкость и время проектирования, что в конечном итоге снижает себестоимость этапа разработки КД и ПД на изделие в целом.

### Список использованных источников

1. Бычков И., Ващук Ю. Конструкторская спецификация – информационная основа управления предприятием // САПР и графика. 2001. № 9. С. 90–95.
2. Леоновец С.А., Гурьянов А.В., Шукалов А.В., Жаринов И.О. Программное обеспечение для автоматизации подготовки текстовой конструкторской документации на программно-управляемые изделия // Программная инженерия - 2017. - Т. 8. - № 3. - С. 129–135.
3. Гатчин Ю.А., Жаринов И.О., Жаринов О.О. Архитектура программного обеспечения автоматизированного рабочего места разработчика бортового авиационного оборудования // Научно-технический информационных технологий, механики и оптики. 2012. № 2. С. 140–141.
4. Гурьянов А.В., Шукалов А.В., Жаринов И.О., Леоновец С.А., Диденко Е.Ю. Управление инженерными данными проекта при автоматизации подготовки конструкторской документации // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 1. С. 182–186 с.