

Altium Designer 14.

Управление жизненным циклом и ревизиями компонентов

Vault, или хранилище, является новым универсальным способом хранения любых данных, проектов и файлов. Наиболее перспективно использовать его в качестве базы данных для библиотечных компонентов. При этом все элементы из Vault можно отобразить на обычной панели Libraries, что удобно для простых пользователей.

В данной статье будет раскрыт смысл понятий жизненного цикла компонента и его различных ревизий, а также описаны способы настройки жизненного цикла и приведены их примеры для библиотек топологических посадочных мест (ТПМ, или PRT), условных графических изображений (УГО, или SYM) и самих компонентов (CMP). Будут показаны примеры создания компонентов.

**Владимир Пранович,
к. т. н.**

v.pranovich@gmail.com

Введение

Непосредственно в Vault хранятся не только компонент и его составные части, но и вся история их создания, модификаций и ревизий. Любая запись в Vault имеет уникальный номер и обозначается Item. Любое новое изменение компонента или любой части в Vault создает очередную запись (Item), при этом образуется новая ревизия. Вы можете не только посмотреть, но и использовать любую ревизию и самого компонента, и его моделей (УГО и ТПМ). Каждая ревизия компонента (модели) может находиться на той или иной стадии жизненного цикла.

Для ревизий необходимо создать схему учета. Она может состоять из одного, двух или трех уровней. Чем больше уровней, тем более разветвленную схему можно сформировать. Здесь мы будем рассматривать только одноуровневую схему, где новая ревизия получает следующий порядковый номер. Новая ревизия образуется всегда, если в библиотечный элемент вносится изменение. Имейте в виду, что компонент содержит модели, и если в модель внесено изменение, то и компонент автоматически получит новую ревизию. Именно поэтому наиболее тщательно относитесь к созданию моделей.

Ознакомиться с общими вопросами жизненного цикла и ревизий компонентов, а также видом их отображения можно в [1, 2]. Ниже представим этап создания конкретного примера жизненного цикла.

Жизненный цикл компонента или его модели может состоять из одного или нескольких этапов, например:

- Этап разработки. Здесь работают только библиотекари, создаются составляющие компонента и определяются его основные параметры, а также производится проверка созданных элементов теми или иными уполномоченными лицами. На данной стадии компоненты еще могут не в полном объеме соответствовать их описаниям. На этом

этапе пользователи могут применять компоненты только с целью их проверки или для временного применения.

- Этап ограниченного использования. На этом этапе можно пользоваться компонентами только при разработке новых изделий или с разрешения вышестоящего руководства. На этом этапе проверяется доступность изделий, соответствие категории цена/качество и при необходимости применяемые изделия вносятся в список.

- Этап полного использования. Если компонент находится на данном этапе, то рекомендуется его использовать без ограничений.

Отметим, что не все перечисленные этапы могут относиться к моделям. В свою очередь каждый этап может состоять из одной или нескольких стадий. Например, этап разработки включает стадии:

- Стадия создания. Здесь работают только библиотекари и формируются составляющие компонента, определяются его параметры и т. п. На данной стадии компоненты еще могут не в полном объеме соответствовать их описаниям. Другие пользователи не должны применять компоненты, но имеют возможность указывать на замеченные несоответствия.

- Стадия проверки. Компонент и его составляющие проверяются разработчиками, дизайнерами, технологами и представителями других служб, которые могут использовать данный компонент. В частности, контроль может состоять из:

- проверки библиотекарем-администратором соответствия изображений УГО единым требованиям предприятия;
- проверки разработчиками схем на удобство использования УГО на схеме и соответствия его описанию;
- проверки изображений УГО на соответствие стандартам представителями нормоконтроля;
- проверки PRT на соответствие стандартам IPC (или иным стандартам) дизайнерами PCB;

- проверки PRT на соответствие единым требованиям предприятия библиотекарем-администратором;
- проверки соответствия подключенных 3D-моделей. Проверка 2D-графики на предмет соответствия изображения требованиям стандартов. Контроль производится конструктором-механиком;
- проверки параметров CMP на соответствие описаниям компонентов и единым требованиям предприятия представителями технических служб.

Каждая стадия в Vault может иметь собственную цветовую гамму, и это облегчает специалистам выбор компонента для использования в своих проектах. Библиотекарям и проверяющим лицам разрешен доступ по переводу компонента из одной стадии в другую. Переходы между стадиями настраиваются индивидуально.

Ревизии и жизненный цикл

Выбрать схему учета ревизий и определить стадии жизненного цикла можно в свойствах Vault.

Используя команду DXP → Preference → Data Management → Vaults, откройте окно подключения Vault. Наведите указатель мыши на название Vault, а затем на ревизию или жизненный цикл, которые вы хотите изменить, и нажатием правой кнопки вызовите контекстное меню (рис. 1).

- Для определения способа учета ревизий следует выбрать пункт меню: **Edit Revision Naming Schemes...**
- Для определения схемы жизненного цикла и его стадий следует выбрать пункт меню: **Edit Lifecycle Definitions...**

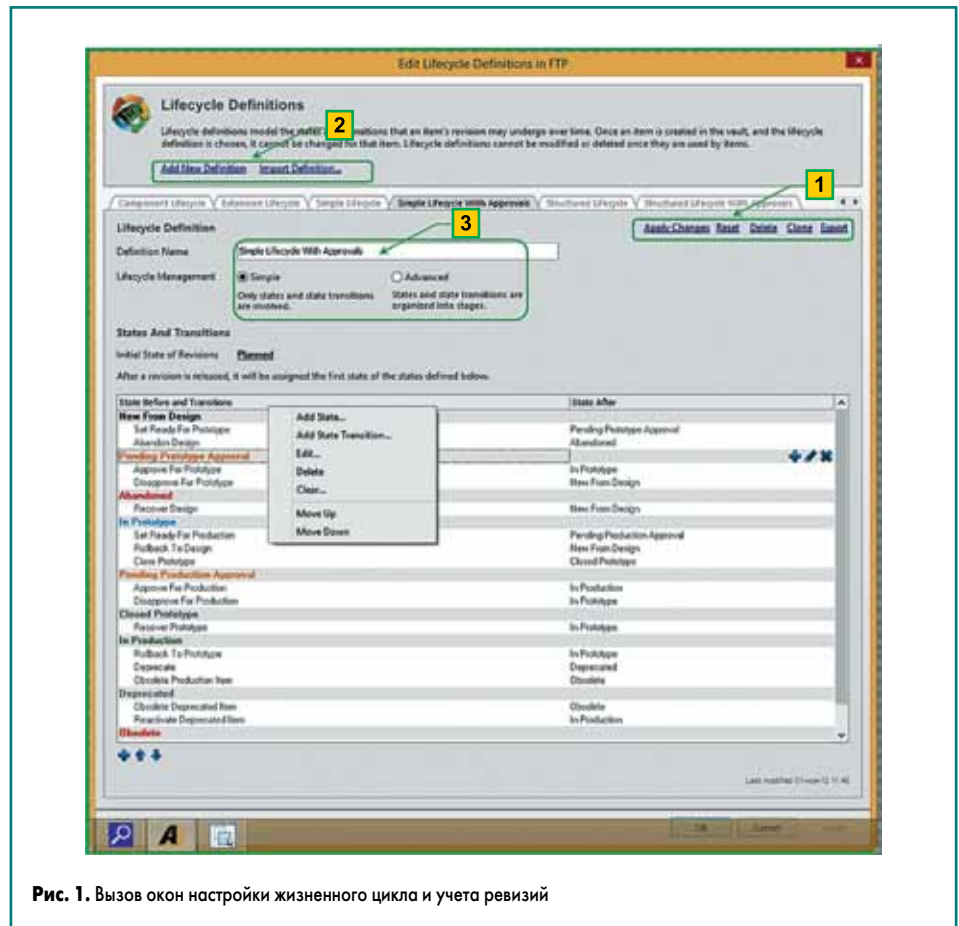


Рис. 1. Вызов окон настройки жизненного цикла и учета ревизий

Ревизии и учет компонентов

В процессе работы над созданием компонента и его составляющих всегда возможно состояние, когда следует внести изменения. Каждому компоненту или его составляющим, а также их ревизиям, размещаемым в Vault,

присваиваются уникальные номера. Учет ревизий выполняется через систему суффиксов, и порядок их присвоения определяется выбранной схемой учета ревизий. Схема учета может быть одно-, двух- и трехуровневой. Вы вправе выбрать любую из предлагаемых схем ревизий или создать свою.

Например, одноуровневую систему применяют для моделей компонента, свойства которых не меняются с течением времени. Такой моделью является топологическое посадочное место для компонентов. Двухуровневую — тогда, когда свойства модели зависят не только от описания компонента, но и от предпочтений разработчиков, например условные графические отображения (чей вид может изменяться в пределах стандарта предприятия). Трехуровневую — для самих компонентов, данные которых могут изменяться (смена поставщика и т. п.).

Покажем пример создания трехуровневой схемы ревизий (рис. 2):

1. Поле **Revision ID Levels**. Здесь укажем количество уровней для уникального (ID) номера, например 3.
2. Поле **Item and Revision Separator**. Выберите один из доступных разделителей в ID, который будет отделять название компонента от обозначения его ревизии.
3. Поля **Caption**. Описательная часть назначения уровней. Здесь укажите назначение уровня, а также символ разделителя. Символ разделителя можно определить индивидуально для каждого уровня. Пример ввода полей и разделителей может выглядеть так:
 - **Packing** — верхний уровень, вариант поставки компонентов в различной упаковке; разделитель не используем.

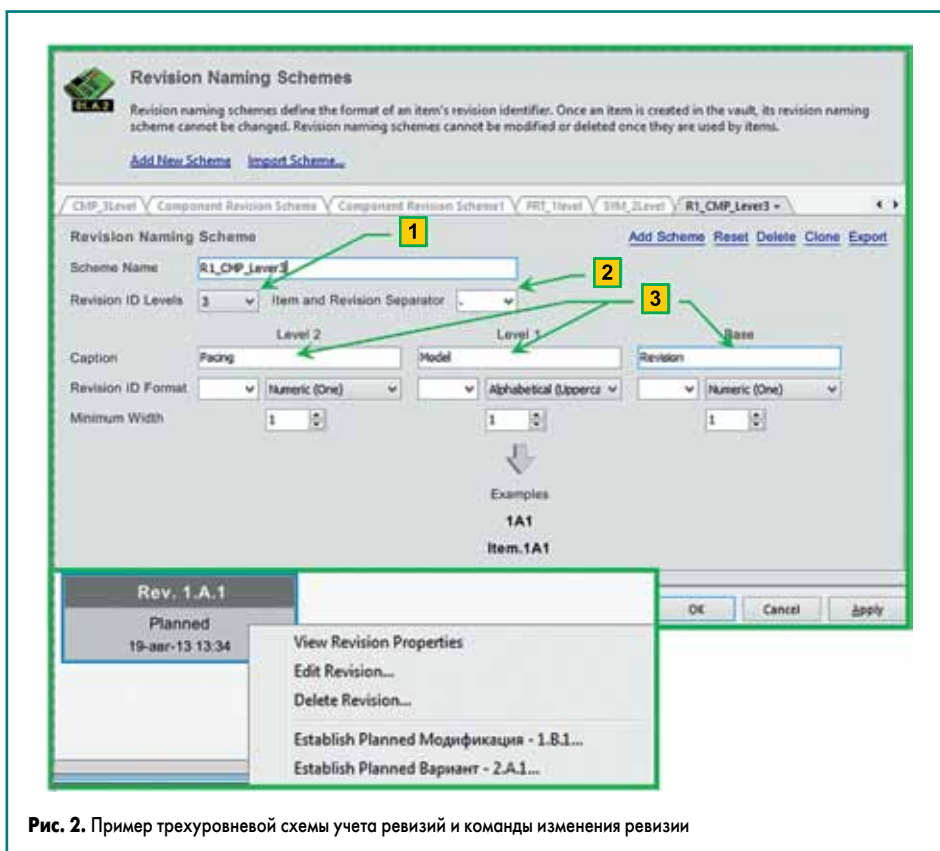


Рис. 2. Пример трехуровневой схемы учета ревизий и команды изменения ревизии

- **Model** — средний уровень, варианты применения моделей; разделитель — точка.
- **Revision** — ревизия компонента или модели; разделитель — точка.
- Поле **Minimum Width** — указываем количество знаков (в нашем примере «1» для всех уровней) представления обозначения номера уровня.

Изменение уровня ревизии компонента производится через контекстное меню. Вызов — нажатие ПКМ по полю отображения ревизии (рис. 2). Для изменения уровня ревизии нужно выбрать одну из доступных команд **Establish...** Команда выглядит так: **Establish** (Planned — состояние компонента из жизненного цикла) (Модификация — название уровня ревизии, который будет повышен) — (XX.XX — значение следующего уровня ревизии, которое будет присвоено). Примечание: базовый уровень ревизий автоматически изменяется при каждом изменении компонента.

Определение жизненного цикла

В окне настроек жизненного цикла уже существуют готовые к использованию примеры. Вы вправе выбрать любой из предлагаемых жизненных циклов или создать и добавить собственный. Можно использовать предлагаемые по умолчанию варианты жизненного цикла. Однако если у вас нет опыта, множество их разновидностей затрудняет с первого раза выбор оптимального жизненного цикла. Мы подробно рассмотрим процесс создания собственного жизненного цикла, и, надеюсь, это поможет вам сориентироваться в вопросе выбора существующего или создать собственный жизненный цикл.

Скорость разработки и внедрения проектов — основной залог успеха. Поэтому главный принцип при создании жизненного цикла таков: на любом этапе цикла пользователь вправе применить тот или иной компонент (модель). Итогом всего жизненного цикла является правильно созданный элемент, который можно использовать без ограничений (если они не введены специально).

Предложим вашему вниманию три типа жизненного цикла: для библиотечных компонентов PRT, SYM, CMP.

Операции с жизненным циклом

Итак, откроем, как указано выше, окно **Lifecycle Definitions** (рис. 4). Там вы найдете встроенные жизненные циклы.

С любым жизненным циклом вы можете:

- **Apply Changes** — провести любые изменения и сохранить их;
- **Reset** — восстановить изменения;
- **Delete** — удалить жизненный цикл;
- **Clone** — использовать любой жизненный цикл для создания другого;
- **Export** — экспортировать жизненный цикл для применения другими.

Вы всегда можете добавить новый или импортировать сохраненный жизненный цикл, используя команды **Add New Definition** или **Import Definition...**

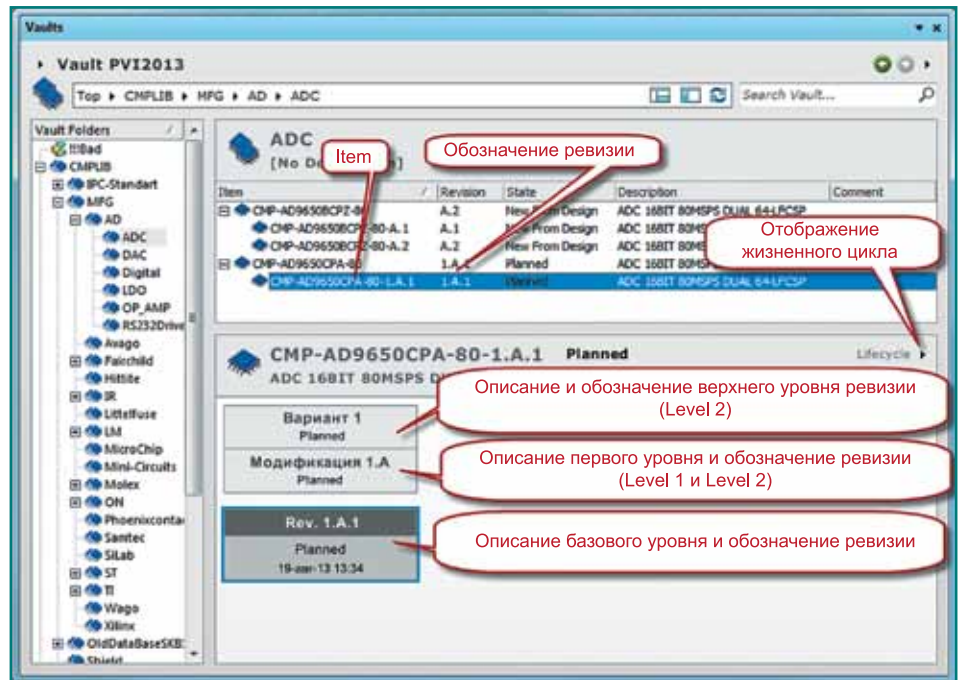


Рис. 3. Отображение компонента в окне Vault в режиме представления жизненного цикла

У вас есть возможность дать название новому жизненному циклу или изменить название существующего, а также выбрать его структуру. Поддерживается как простой (Simple), так и сложные (Advanced) жизненные циклы, которые могут состоять из одной, двух или до трех стадий.

Рассмотрим пример простого жизненного цикла (**Simple Life With Approvals**) и его настройку. Если этот цикл уже используется, он не может быть изменен. Однако есть возможность клонировать его и, скажем, доба-

вить стадии и переходы между ними, а также привести названия этих состояний, стадий и самих жизненных циклов на русском языке. Последнее не рекомендуется делать, если ваш дизайн проектов предусматривает работу не только на территории СНГ.

Итак, в строке Definition Name введем новое название жизненного цикла «1PRT1», что будет означать 1 вариант жизненного цикла из одной стадии, предназначенный для топологического посадочного места. Название вкладки автоматически изменится, а в его

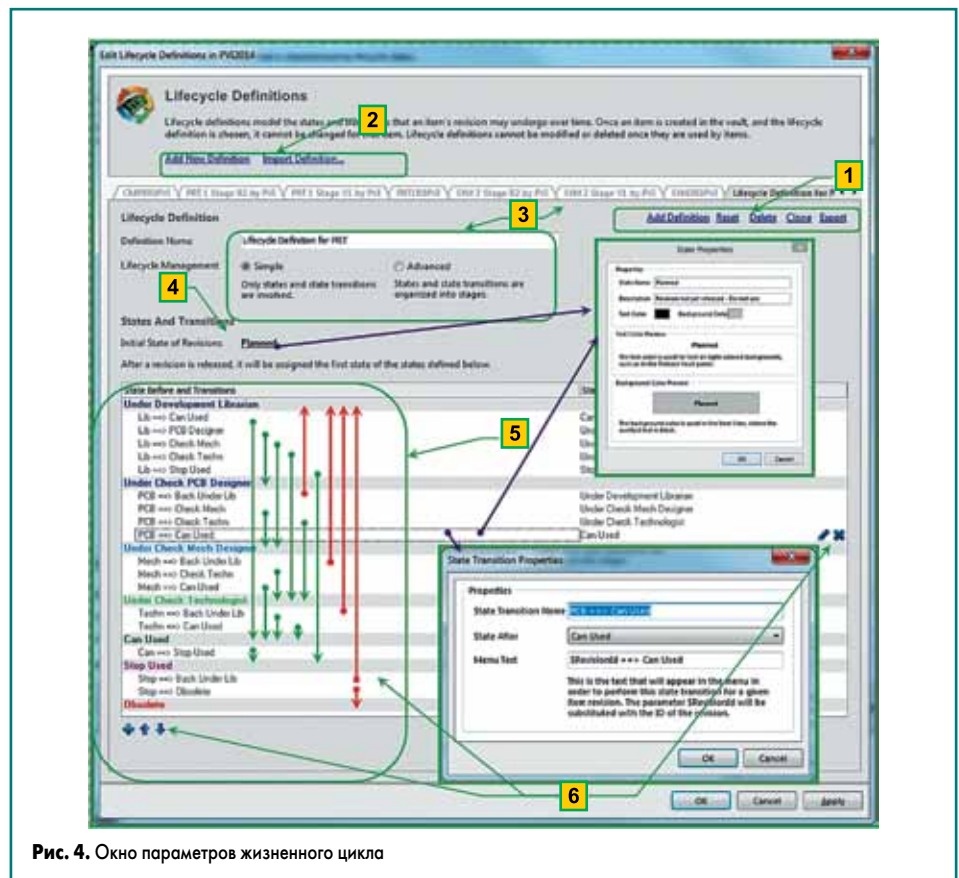


Рис. 4. Окно параметров жизненного цикла

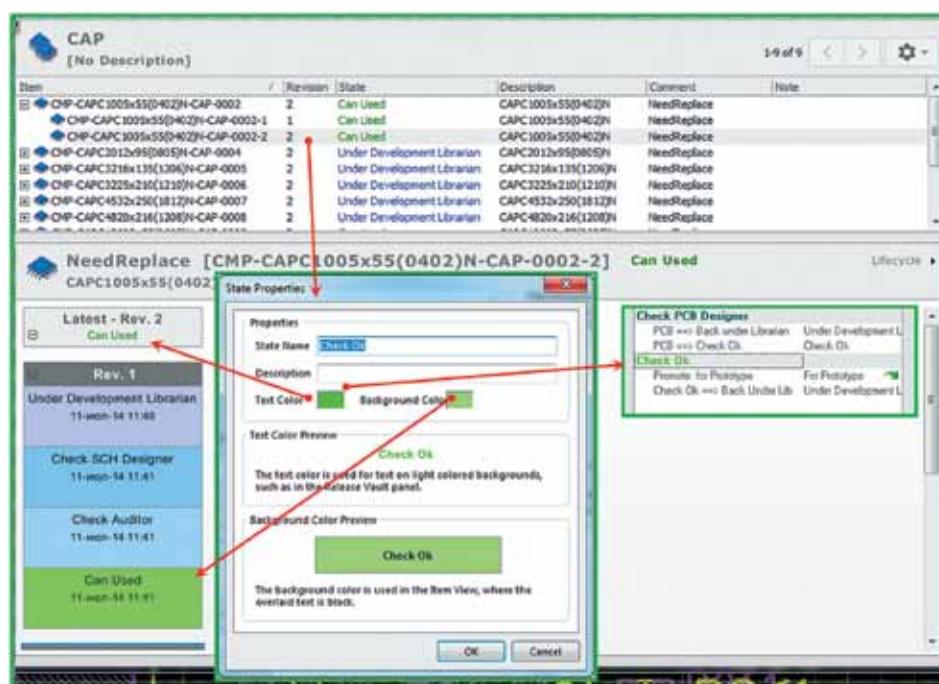


Рис. 5. Цветовая гамма жизненного цикла в браузере Vault

конец появится звездочка, это означает, что жизненный цикл изменен, но не сохранен.

Initial State Revision. Жизненный цикл компонента или его составной части, а также любого элемента в Vault будет начинаться с данного состояния. Это состояние всегда присутствует в жизненном цикле и не может быть удалено. У вас есть возможность изменить название нажатием ЛКМ на надписи **Planned**. Обращаю внимание: вы можете настроить цвет и фон надписи. Именно этим цветом и фоном будет обозначаться начальное состояние жизненного цикла компонента в Vault (рис. 5). Отмечу также, что подобную цветовую гамму вы можете настроить для каждой стадии и состояния в стадии жизненного цикла. Состояние **Planned** получается при резервировании Item для топологического места непосредственно через браузер Vault и не содержит ни одной ревизии.

Обязательно выберите и настройте цветовую гамму не только для стадий жизненного цикла, но и всех его состояний. Это существенно облегчит отображение состояния компонента (модели) в браузере Vault.

Каждая стадия жизненного цикла имеет несколько состояний (обозначены цветными надписями). Вы также должны указать переходы между состояниями (серые надписи) и установить порядок их следования друг за другом. Добавление, перемещение и редакция осуществляются с помощью стрелок и других знаков, указанных на рисунке. При настройке переходов обратите особое внимание:

- Поле **State Transition Name** — уникальное (в пределах всего жизненного цикла) название перехода. Именно оно будет отражаться в окне как название перехода.
- Поле **State After** — здесь следует выбрать одно или несколько из доступных состояний (**State Transition Name**), куда можно перейти из текущего состояния.

- Поле **Menu Text** — здесь нужно указать надпись меню, которая будет предложена в браузере Vault, для перехода в новое состояние. Для указания названия ревизии компонента можете использовать ссылку **\$RevisionItem**.

В момент создания новых состояний и переходов вы не всегда можете указать переход в нужное состояние, так как оно еще не сформировано. Не беспокойтесь и указывайте любое из существующих. Когда все состояния жизненного цикла будут созданы, вы легко настроите нужный переход.

Пример создания жизненного цикла топологического посадочного места (PRT)

Рассмотрим реализацию наиболее простого, одноуровневого цикла для топологического посадочного места (PRT). Главная цель, к которой мы будем стремиться при создании жизненного цикла, — получить в итоге корректное посадочное место. Основопологающим принципом при работе с библиотеками (в том числе PRT) станет то, что любые изменения (создание ревизий) может производить только библиотекарь, а переходы между состояниями жизненного цикла выполняет как библиотекарь, так и контролирующее на данном этапе лицо. В этом случае единообразие всех создаваемых компонентов будет поддерживать библиотекарь.

Предусмотрим в жизненном цикле, состоящем из одной стадии, следующие состояния:

1. Этап создания PRT библиотекарем (обязательный).
2. Этап проверки дизайнером PCB (при необходимости).
3. Этап проверки механиком-конструктором (при необходимости).

4. Этап проверки технологом (при необходимости).
5. Разрешение на использование PRT (обязательный).
6. Запрет применения PRT по тем или иным причинам (обязательный).
7. Завершение жизненного цикла (обязательный).

На рис. 4 указана реализация данного жизненного цикла. Жизненный цикл содержит следующие стадии и переходы:

1. **Under Development Librarian**. Эта стадия работы библиотекаря означает, что посадочное место находится в стадии разработки. В зависимости от типа и назначения посадочного места библиотекарь может изменить на любую из приведенных ниже стадий.
2. **Under Check PCB Designer**. Это состояние проверки созданного ТПМ PCB-дизайнером. Проверке подлежат:
 - Наименование и положение PAD.
 - Параметры PAD.
 - Параметры и графика на всех слоях.
 - Подключенные 3D-модели и параметры простых 3D-тел.
 При обнаружении несоответствия или необходимости доработки производится возврат в состояние **Under development Librarian**. Если замечаний нет, в зависимости от типа и назначения посадочного места происходит переход на одну из следующих стадий.
3. **Under Check Mech Designer**. Это стадия проверки механиком-конструктором. На данном этапе выполняется проверка графики ТПМ на тех механических слоях, где изображена графика компонента для сборочного чертежа. Здесь проверяются также все параметры 3D-моделей и параметры простых 3D-тел, а также формируется 3D-модель для MCAD при передаче через IDF-формат.

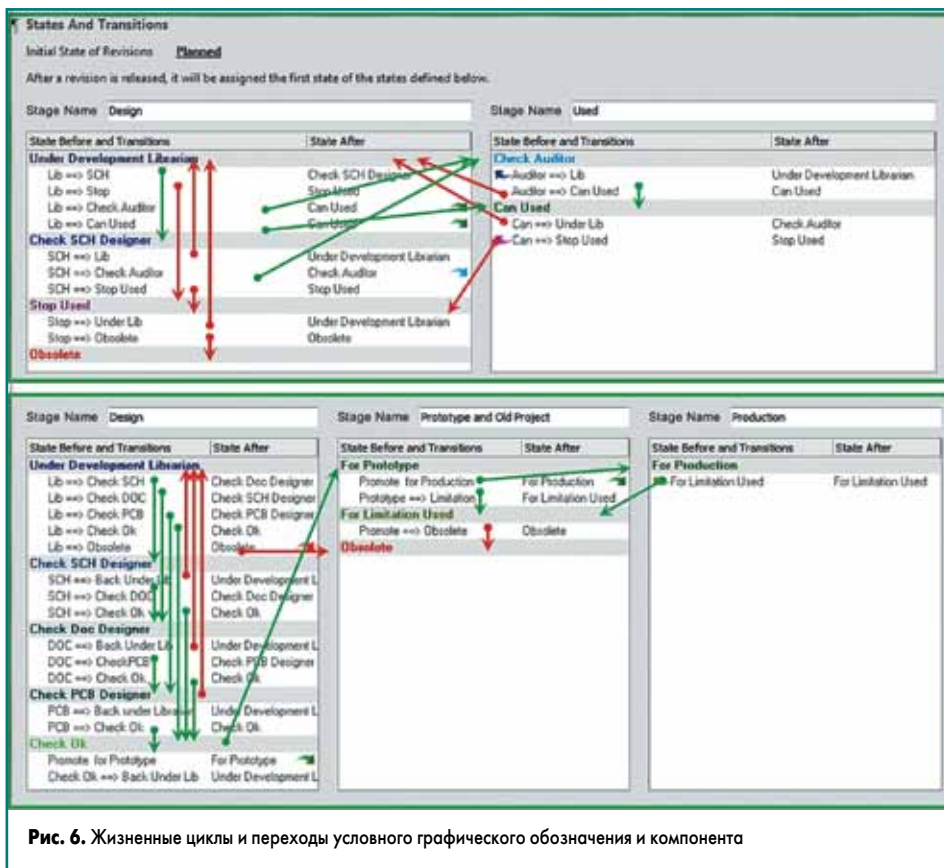


Рис. 6. Жизненные циклы и переходы условного графического обозначения и компонента

ходы между состояниями, если при создании и проверке УГО ошибки не обнаружены, а красными — где требуется доработка УГО или установлен запрет на его использование.

Примеры жизненного цикла компонента

Особенности компонента в том, что помимо параметров он также содержит модели, которые в свою очередь являются элементами Vault. Кроме этого, компоненты могут быть сняты с производства или запрещены к применению в той или иной мере. Предложим для компонента трехуровневый жизненный цикл.

Жизненный цикл компонента состоит из трех стадий: разработка (Design), ограниченное применение (Prototype and Old Project), для производства (Production) — и девяти состояний (рис. 6):

I. Стадия 1. Разработка компонента. Цикл определения параметров и указания моделей для компонента:

1. **Under Development Librarian** — создание компонента библиотекарем, ввод параметров и моделей.
2. **Check SCH Designer** — проверка дизайнерами схем и PCB подключенных моделей и вспомогательными службами правильности заполнения параметров.
3. **Check Doc Designer** — проверка полноты и значений параметров.
4. **Check PCB Designer** — проверка дизайнером PCB подключенных моделей.
5. **Check Ok** — успешное создание и возможность первичного использования.

II. Стадия 2. Ограниченное использование компонента:

6. **For Prototype** — использование СМР для разовых заказов или первичное использование в новой разработке.
7. **For Limitation Used** — использование СМР для разовых заказов или поддержки завершенных разработок.
8. **Obsolete** — запрет использования.

III. Стадия 3. Для применения при выпуске продукции:

9. **For Production** — использование для выпуска продукции.

Заключение

Отметим, что переход компонента из одной стадии в другую должен проходить через создание новой ревизии, если требуется дополнение его свойств данными бухгалтерского и складского учета, номерами разрешительных или запретительных списков и иной необходимой информацией.

Литература

1. Vault. Управление жизненным циклом и ревизиями компонентов. <http://wiki.altium.com/pages/viewpage.action?pageId=45638895>
2. Библиотека компонентов CMPLIB. <http://wiki.altium.com/pages/viewpage.action?pageId=44957741>

При обнаружении несоответствия или необходимости доработки производится возврат в состояние **Under Development Librarian**. Если замечаний нет, в зависимости от типа и назначения посадочного места переход на одну из следующих стадий.

4. **Under Check Technologist**. Проверка зон запрета для соседних компонентов, размеров площадок и т.п. осуществляется на соответствие требованиям технологии сборки печатного узла. Проверка осуществляется технологом. Проверка может считаться пройденной, если по результатам сборки печатных узлов не превышен процент допустимого брака при установке данного компонента.

При обнаружении несоответствия или высоком браке при монтаже производится возврат в состояние **Under Development Librarian** для доработки посадочного места. В противном случае — переход на стадию **Can Used**.

5. **Can Used**. Это возможность использования ТПМ всеми, кому открыт доступ.
6. **Stop Used**. Временная приостановка на использование ТПМ по тем или иным причинам. При решении возобновить использование — возврат к разработке библиотекарем. В противном случае — переход на стадию **Obsolete**.
7. **Obsolete**. Полный запрет на применение данного посадочного места.

На рис. 4 зелеными стрелками указаны переходы, если при проверках не обнаружено ошибок, красными — если требуется доработка. На рис. 5 представлена цветовая гамма отображения вида ревизий и состояний жизненного цикла и стрелками указано влияние настроек цветовой гаммы состояний.

Пример жизненного цикла условного графического обозначения

Условное графическое обозначение отличается тем, что его вид может быть разным и объем предоставления информации тоже. Изображение УГО должно соответствовать его описанию и уже на этом этапе может применяться в разработке. Второе условие — соблюсти общепринятое изображение или удовлетворить требованиям ГОСТов или иных стандартов. Вопросы согласования изображений могут занимать длительное время, но это не должно затрагивать разработку. Предложим для этого двухуровневый жизненный цикл.

Цикл состоит из двух стадий: разработка (Design) и применение (Used) и шести состояний.

- Стадия разработки включает состояния:
1. **Under Development Librarian** — этап создания УГО библиотекарем (обязательный).
 2. **Check SCH Designer** — этап проверки дизайнером соответствия УГО технической документации и удобства использования на схеме (при необходимости).
 3. **Stop Used** — запрет применения УГО по тем или иным причинам (обязательный).
 4. **Obsolete** — завершение жизненного цикла УГО (обязательный).

Стадия применения предусматривает состояния:

5. **Check Auditor** — проверка соответствия УГО нормативным документам (нормоконтроль).
6. **Can Used** — применение УГО для использования в компонентах.

На рис. 6 представлен данный жизненный цикл, где зелеными стрелками указаны пере-