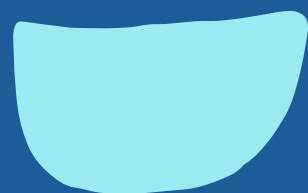


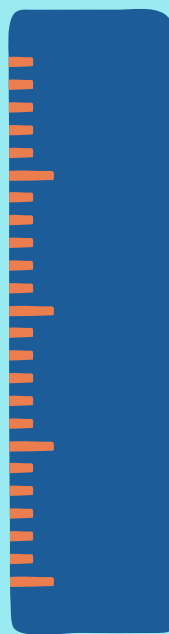
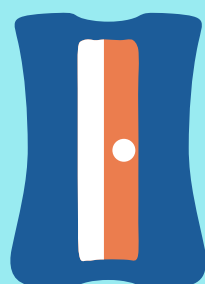
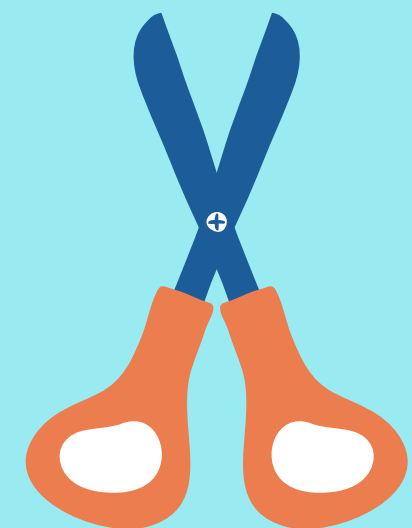
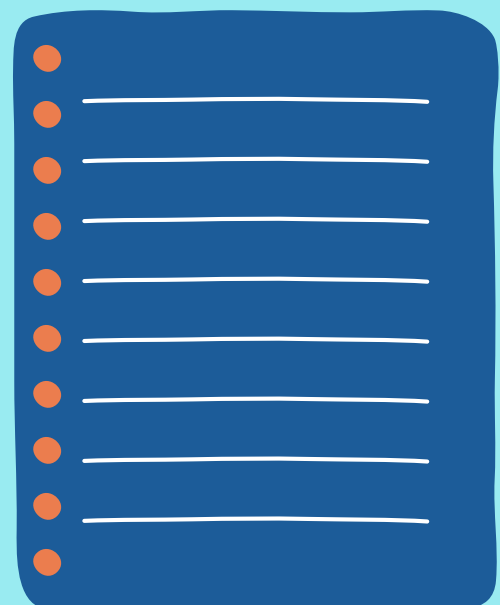
Лекция 4

УПРАВЛЕНИЕ IT- ПРОЕКТАМИ

Изотова Анастасия Андреевна



Управление риском



Риск – возможность опасности, неудачи.
Влияние риска вычисляют по выражению:

$RE = P(UO) \times L(UO)$, где

RE – показатель риска (Risk Exposure – подверженность риску);

$P(UO)$ – вероятность неудовлетворительного результата;

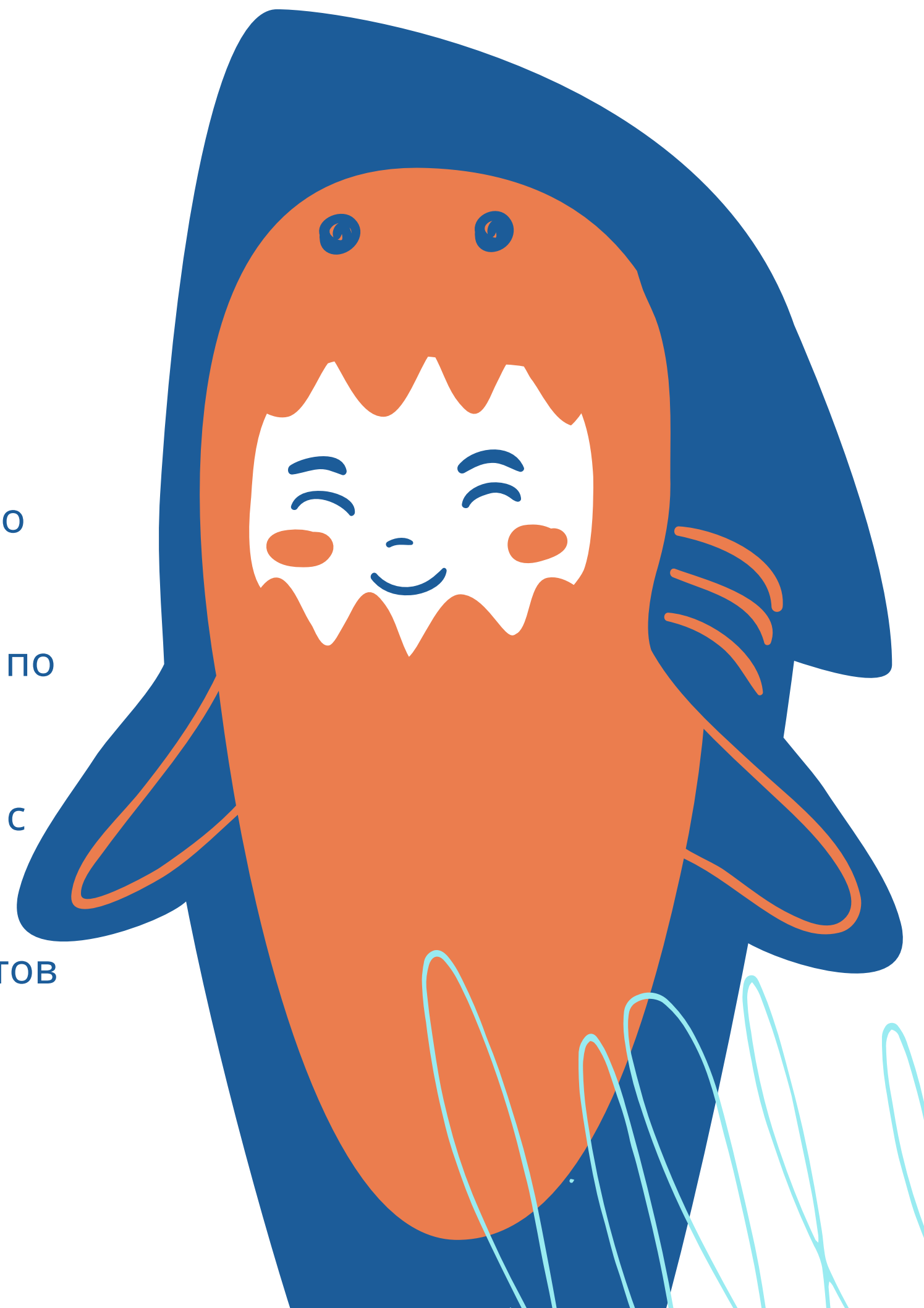
$L(UO)$ – потеря при неудовлетворительном результате;

Неудовлетворительным результатом может быть превышение бюджета, низкая надежность, неправильное функционирование и т.д.

Управление риском

Управление риском включает:

- оценка риска**
 - 1 Идентификация риска – выявление элементов риска в проекте.
 - 2 Анализ риска – оценка вероятности и величины потери по каждому элементу риска.
 - 3 Ранжирование риска – упорядочивание элементов риска по степени их влияния.
- контроль риска**
 - 4 Планирование управления риском – подготовка к работе с каждым элементом риска.
 - 5 Разрешение риска – устранение или разрешение элементов риска.
 - 6 Наблюдение риска – отслеживание динамики элементов риска, выполнение корректирующих действий.



ИДЕНТИФИКАЦИЯ РИСКА

Категории источников

риска:

- проектный риск;
- технический риск;
- коммерческий риск.



Источники проектного

риска:

- выбор бюджета, плана, человеческих ресурсов программного проекта;
- формирование требований к программному продукту;
- сложность, размер и структура программного проекта;
- методика взаимодействия с заказчиком.



Идентификация риска

Источники технического риска:

- трудности проектирования, конструирования, формирования интерфейса, тестирования и сопровождения;
- неточность спецификации;
- техническая неопределенность или отсталость принятого решения.

Источники коммерческого риска:

- создание продукта, не требующегося на рынке;
- создание продукта, опережающего требования рынка (отстающего от них);
- потеря финансирования.

Анализ риска



Элемент риска	Вероятность, %	Потери	Влияние риска
1. Критическая программная ошибка	3–5	10	30–50
2. Ошибка потери ключевых данных	3–5	8	24–40
3. Отказоустойчивость недопустимо снижает производительность	4–8	7	28–56
4. Отслеживание опасного условия как безопасного	5	9	45
5. Отслеживание безопасного условия как опасного	5	3	15
6. Аппаратные задержки срывают планирование	6	4	24
7. Ошибки преобразования данных приводят к избыточным вычислениям	8	1	8
8. Слабый интерфейс пользователя снижает эффективность работы	6	5	30
9. Дефицит процессорной памяти	1	7	7
10. СУБД теряет данные	2	2	4

Оценка влияния элементов риска

УПРАВЛЕНИЕ ПЕРСОНАЛОМ

Подбор членов команды (учитываются следующие аспекты):

- 1) опыт работы во многих аппаратно-программных средах;
- 2) знание языков программирования;
- 3) образование и опыт работы по специальности;
- 4) коммуникабельность;
- 5) способность адаптироваться;
- 6) жизненная позиция;
- 7) личностные качества.

Преимущества разработки без персонализации:

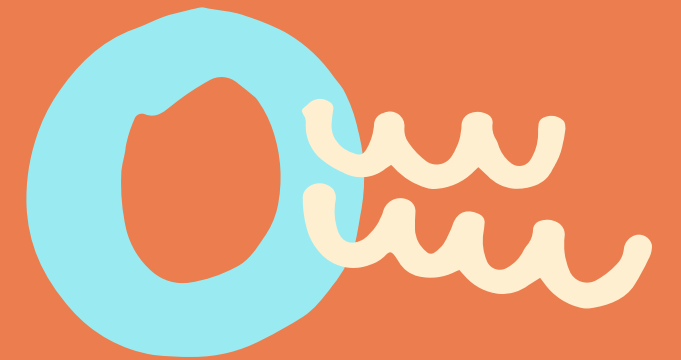
- упрощение процедур проверки, критики недостатков, повышение их объективности;
- поощрение стиля непринужденного обсуждения рабочих заданий, достоинств и недостатков отдельных решений;
- активизация дружеских отношений, повышение уровня искренности;
- быстрый рост мастерства (благодаря работы бок о бок);
- улучшение качества, совершенствования результатов работы.



Управление персоналом

Параметры, влияющие на эффективность взаимодействия в команде:

- 1) размер/структура команды;
- 2) иерархия команды;
- 3) рабочее окружение.



Состав группы по типовым ролям:

- 1) Аналитик
- 2) Архитектор
- 3) Конструктор компонентов
- 4) Специалист по применению
- 5) Специалист по повторному использованию
- 6) Специалист по интеграции
- 7) Специалист по документации
- 8) Системный программист
- 9) Системный администратор



Управление документацией

Управление документацией ориентировано на поддержание ее полноты и **согласованности**.

Полнота подразумевает, что комплект документации должен охватывать весь жизненный цикл ПО.

Согласованность означает, что комплект документов не содержит внутренних противоречий.



Организации, отвечающие за стандарты документации:

- 1) Государственный комитет Российской Федерации по стандартизации и метрологии (Госстандарт России).
- 2) Международная организация по стандартизации (ISO) и Международная электротехническая комиссия (IEC).
- 3) Институт инженеров по электротехнике и радиоэлектронике (IEEE).
- 4) Институт программной инженерии (SEI).
- 5) Консорциум по технологии манипулирования объектами (OMG).



УПРАВЛЕНИЕ КОНФИГУРАЦИЕЙ

Управление конфигурацией – это координация различных версий и частей документации и программного кода. Оно является защитной деятельностью, применяемой на всех этапах жизненного цикла ПО, и обеспечивает управление изменениями в ПО, которое включает в себя следующие действия:

- идентификацию изменения;
- контроль изменения;
- гарантию правильной реализации изменения;
- формирование сообщения об изменениях.

За время жизни программный код продукта претерпевает изменения двух категорий: добавление новых частей и подключение новых версий существующих частей.

Информацию о выходе процесса разработки ПО можно разделить на следующие категории:

1. компьютерные программы (в виде исполняемых кодов);
2. документы, описывающие программы (как для технического персонала, так и для пользователей);
3. структуры данных (как внешние, так и внутренние).

Совокупность всех элементов информации, вырабатываемых как часть процесса разработки ПО, называют конфигурацией ПО.

Идентификация объектов в конфигурации ПО

Идентифицирующиеся типы объектов:

- базисные объекты;
- составные объекты.

Базисный объект – элемент информации, создаваемый в ходе анализа, проектирования, кодирования и тестирования.

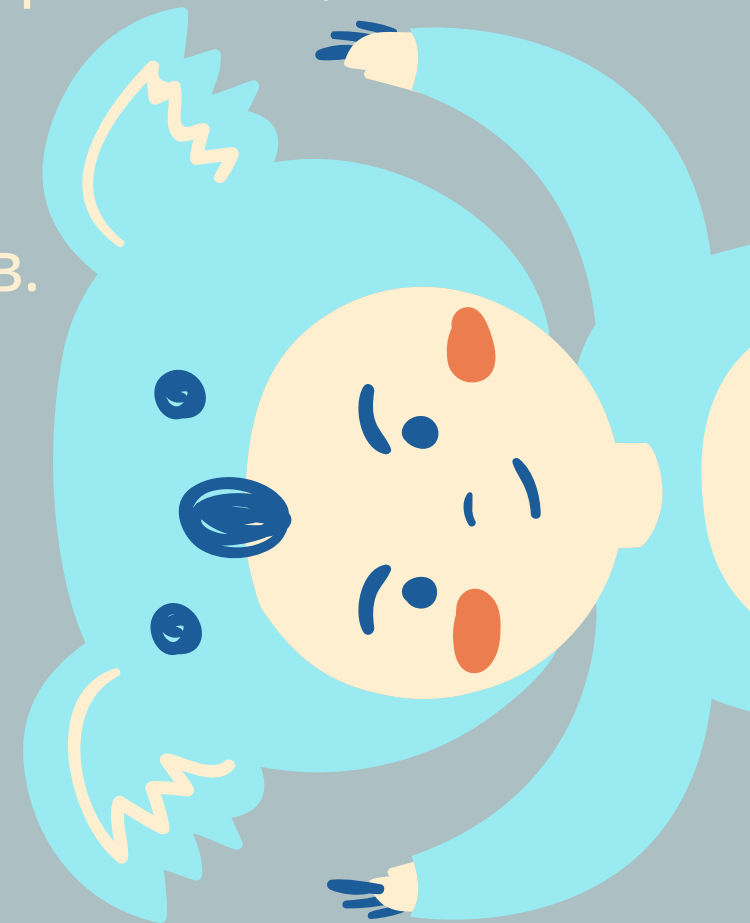
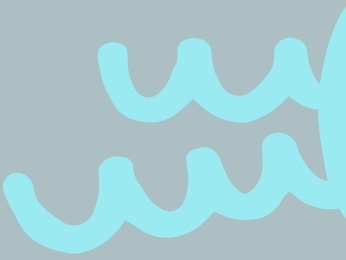
Составной объект – коллекция базисных объектов и других составных объектов.

Описание объекта – перечень элементов данных, определяющий:

- тип элемента (элемент модели, программа, данные);
- идентификатор проекта;
- информацию изменения и/или версии.

Ресурсы – это элементы, которые предоставляются, обрабатываются, указываются или как-то иначе запрашиваются объектом.

Реализация – указатель на элемент текста для базисного объекта и null для составного объекта.



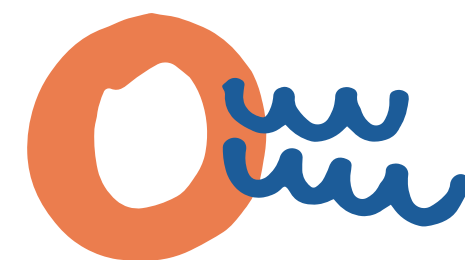
КОНТРОЛЬ ВЕРСИЙ



Контроль версий объединяет процедуры и средства для управления различными версиями объектов конфигурации, которые создаются в ходе разработки.

Система контроля версий обычно состоит из следующих элементов:

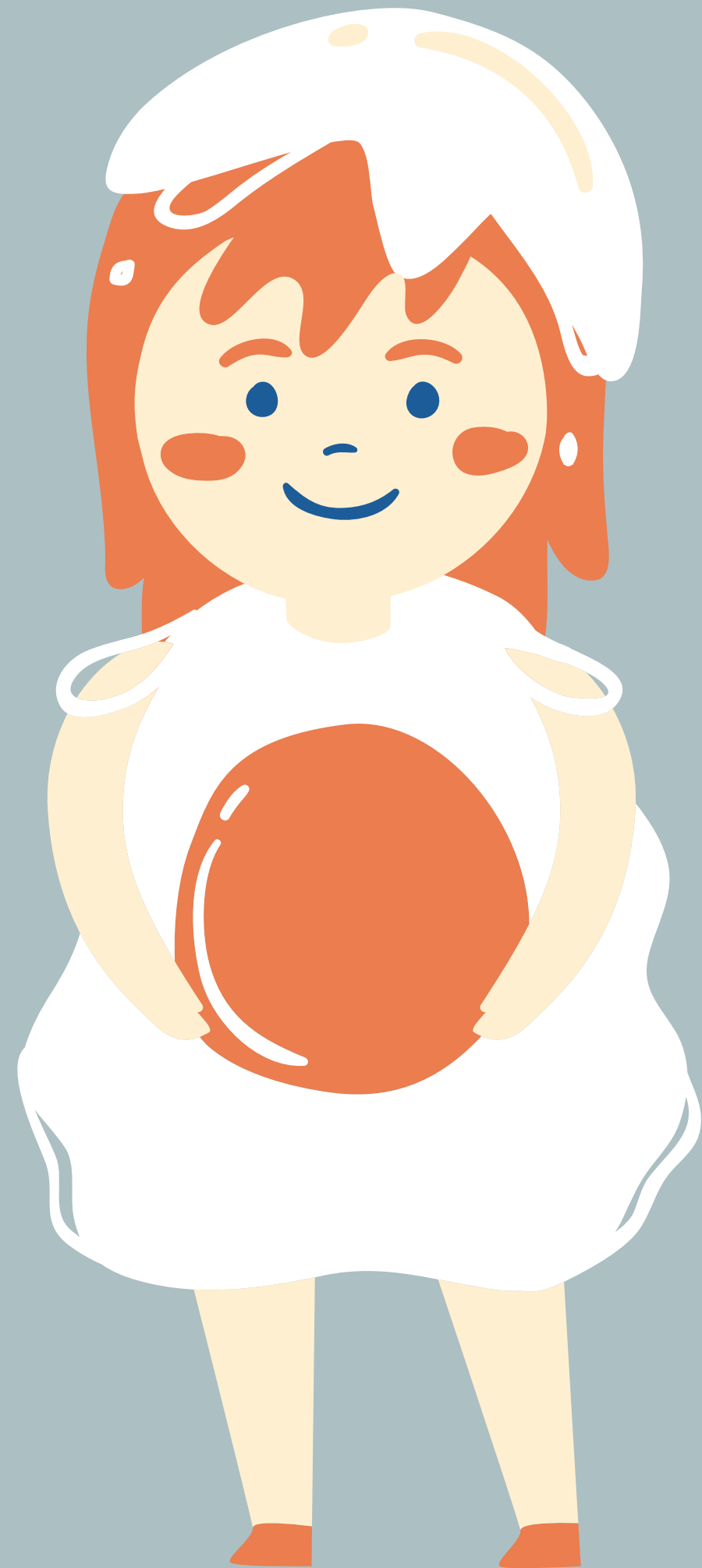
- 1) база данных проекта, сохраняющая все значимые объекты конфигурации;
- 2) средство управления версиями, сохраняющее все версии объектов конфигурации (или создающее любую версию на основе различных предыдущих версий);
- 3) устройство генерации, позволяющее собирать все значимые объекты конфигурации и создавать определенную версию ПО.



ПЛАН УПРАВЛЕНИЯ КОНФИГУРАЦИЕЙ



1. Введение.
2. Управление конфигурациями.
 - 2.1. Организация.
 - 2.2. Ответственность за управление конфигурациями.
 - 2.3. Применяемые методики, директивы, процедуры.
3. Виды деятельности.
 - 3.1. Определение элементов конфигурации.
 - 3.1.1. Именованение элементов конфигурации.
 - 3.1.2. Получение элементов конфигурации.
 - 3.2. Контроль конфигурации.
 - 3.2.1. Запрос на изменения.
 - 3.2.2. Оценка изменений.
 - 3.2.3. Одобрение или неодобрение изменений.
 - 3.2.4. Реализация изменений.
 - 3.3. Определение статуса конфигурации.
 - 3.4. Аудиты и проверки конфигурации.
 - 3.5. Контроль интерфейса.
 - 3.6. Контроль поставщиков и субподрядчиков.
4. Расписание.
5. Ресурсы.
6. Сопровождение.



СПАСИБО
ЗА
ВНИМАНИЕ!

