



Жизненный цикл ПО

лектор: Парамонов А.И.



БГТУ – 2020

ЖИЗНЕННЫЙ ЦИКЛ (life cycle)

- Одним из базовых понятий методологии проектирования ИС является понятие **жизненного цикла** ее программного обеспечения (ЖЦ ПО).
- Методология проектирования *информационных систем* (ИС) описывает процесс создания и сопровождения систем в виде их **жизненного цикла** (ЖЦ), представляя его как некоторую последовательность стадий и выполняемых на них процессов.



Понятие жизненного цикла

В авторитетном словаре программной инженерии **IEEE Std 610.12-90** «IEEE Standard Glossary of Software Engineering Terminology» записано:

Жизненный цикл ПО определяется как период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.



Выделяют следующие этапы

ЖЦ ПО:

1. формирование требований к системе (планирование)
2. проектирование
3. реализация (кодирование)
4. тестирование
5. ввод в действие
6. эксплуатация и сопровождение

* Если последний этап не включают в цикл, то тогда говорят только о **периоде создания ПО**




Для каждого этапа определяются:

- *состав и последовательность выполняемых работ,*
- *получаемые результаты,*
- *методы и средства, необходимые для выполнения работ,*
- *роли и ответственность участников и т.д.*

На каждом этапе ЖЦ создаются специфичные для него **модели**.

Модели формируются рабочими группами команды проекта.



Существует целый ряд стандартов,
регламентирующих ЖЦ ПО,
а в некоторых случаях и процессы разработки

Среди наиболее известных стандартов:

- ГОСТ 34.601-90
- ISO/IEC 12207:1995
- Custom Development Method (**CDM** – методика Oracle)
- Rational Unified Process (**RUP**)
- Microsoft Solution Framework (**MSF**)
- Extreme Programming (**XP**).



ISO/IEC 12207

- ISO – International Organization of Standardization – Международная организация по стандартизации;
- IEC – International Electrotechnical Commission – Международная комиссия по электротехнике

Стандарт определяет структуру ЖЦ, содержащую процессы, действия и задачи, которые должны быть выполнены во время создания ПО.

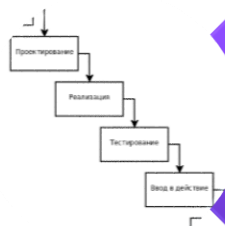


Модель ЖЦ

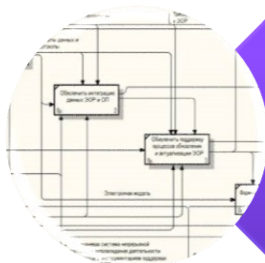
Под моделью ЖЦ понимается структура, определяющая последовательность выполнения и взаимосвязи процессов, действий и задач, выполняемых на протяжении ЖЦ.

Модель ЖЦ зависит от специфики ИС и условий, в которых последняя создается и функционирует

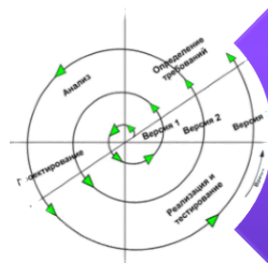
Классические модели ЖЦ



Каскадная модель



Поэтапная модель с промежуточным контролем



Спиральная модель



Водопадная модель (классическая)



Каскадная модель ЖЦ ИС

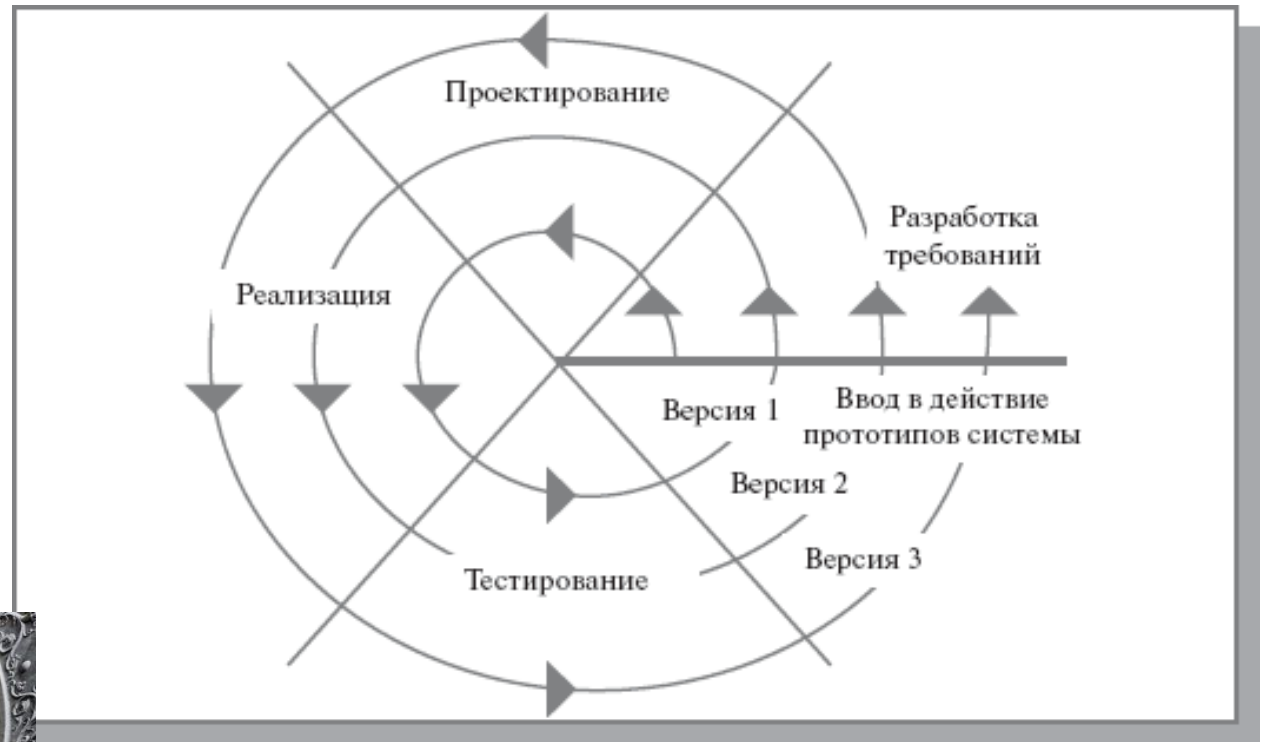


Поэтапная модель с промежуточным контролем



Итерационная модель ЖЦ ИС

Спиральная модель ЖЦ ПО



Методологии проектирования ПО

- Agile software development
- Agile Unified Process (AUP)
- Behavior Driven Development (BDD)
- Big Design Up Front (BDUF)
- C...
- Design-driven development (D3)
- Design Driven Testing (DDT)
- Domain-Driven Design (DDD)
- Dynamic Systems Development Method (DSDM)
- Evolutionary Model
- Extreme Programming (XP)
- Feature Driven Development
- Iterative and incremental development
- Kaizen
- Microsoft Solutions Framework (MSF)
- Model-driven architecture (MDA)
- Open Unified Process
- Rapid application development (RAD)
- Rational Unified Process (RUP)
- Scrum
- Software
- Craftsmanship
- Spiral model
- Structured Systems Analysis and Design
- Test-driven development (TDD)
- Unified Process (UP)
- V-Model
- Waterfall model
- Wheel and spoke model
- ...

Что и Как выбрать ???

Agile Model

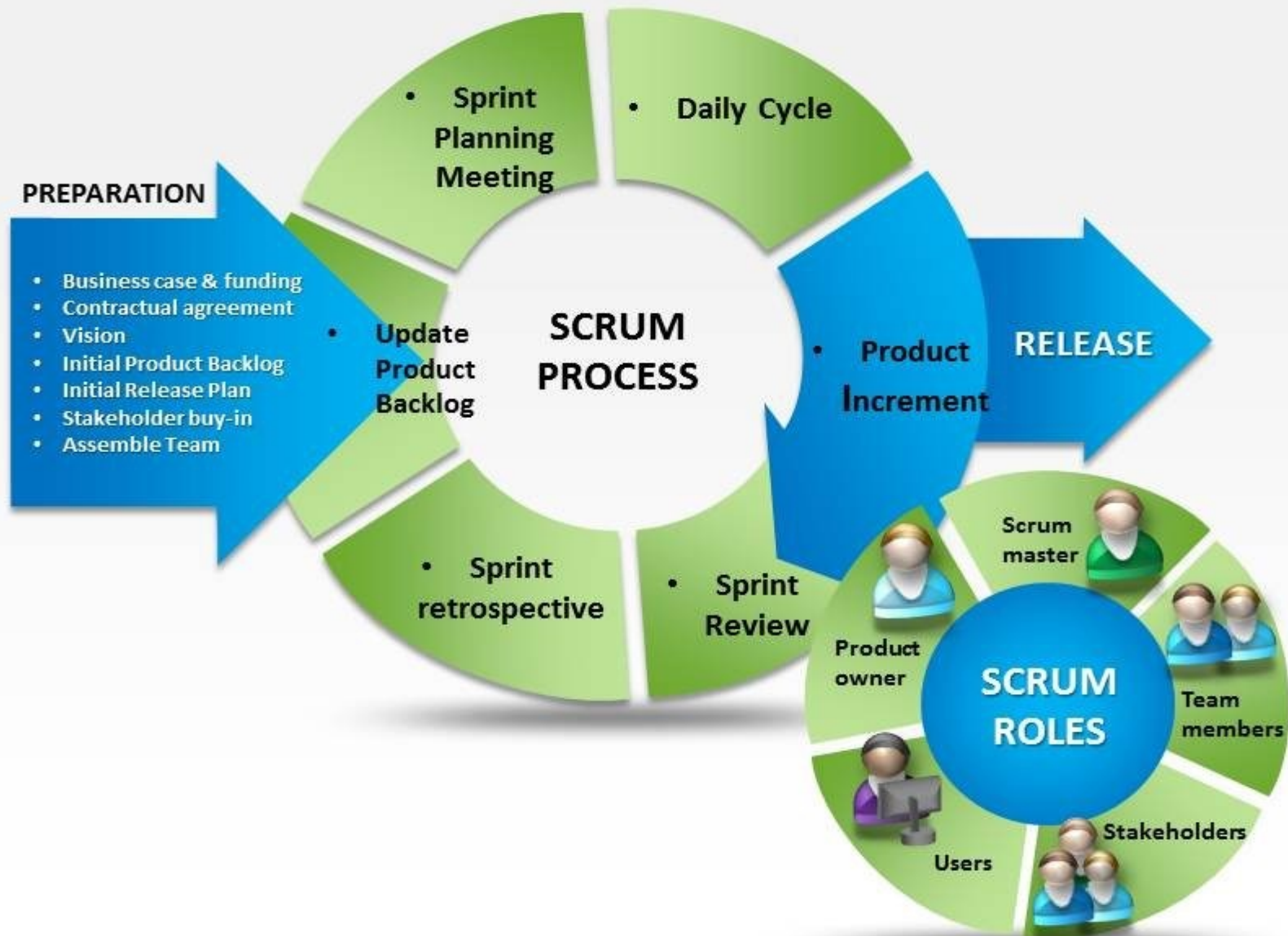
- **Agile** — семейство процессов разработки, а не единственный подход в разработке программного обеспечения, и определяется *Agile Manifesto*.
- Agile не включает практик, а определяет ценности и принципы, которыми руководствуются команды.



SCRUM

- Scrum — это набор принципов, на которых строится процесс разработки, позволяющий в жёстко фиксированные и небольшие по времени итерации, называемые **спринтами** (sprints), предоставлять конечному пользователю работающее ПО с новыми возможностями, для которых определён наибольший приоритет.
- Возможности ПО к реализации в очередном спринте определяются в начале спринта на этапе планирования и не могут изменяться на всём его протяжении. При этом строго фиксированная небольшая длительность спринта придаёт процессу разработки предсказуемость и гибкость.

Scrum Process





The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



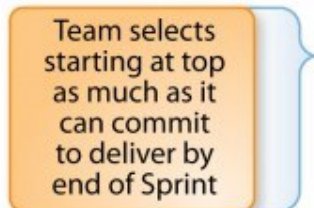
Product Owner



The Team



Product Backlog



Sprint Planning Meeting



Sprint Backlog



1-4 Week Sprint

Sprint end date and team deliverable do not change



Scrum Master



Burndown/up Charts

Every 24 Hours



Daily Scrum Meeting



Sprint Review



Finished Work



Sprint Retrospective



Kanban



Подходы к выбору методологии

- Принцип 1. Большая по размерам методология нужна тогда, когда в проекте занято большое число разработчиков.
- Принцип 2. Большая корректность методологии (видимая со стороны) или, другими словами, "большая плотность" нужна в тех случаях, когда скрытые ошибки в программном продукте могут повлечь за собой значительный ущерб (большая критичность разрабатываемой системы).
- Принцип 3. Незначительное увеличение "размеров" или "плотности" методологии ведет к существенному увеличению стоимости проекта.
- Принцип 4. Наиболее эффективная форма коммуникации (для передачи идей) - непосредственное взаимодействие, лицом к лицу, как при рисовании у доски.

