

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

ВВЕДЕНИЕ В ПРОГРАММНУЮ ИНЖЕНЕРИЮ



Евгений
Александрович
Мирошниченко

mir@tpu.ru

Отделение информационных технологий

Инженерная школа
информационных технологий
и робототехники

Томский политехнический университет

Сайт дисциплины:

vt.tpu.ru → Дисциплины → Технологии программирования

http://portal.tpu.ru/departments/kafedra/vt/Disciplines_VT/SoftwareEngineering

Основная литература:

- *Брукс Ф.* Мифический человеко-месяц или как создаются программные системы. — СПб.: Символ-Плюс, 2010. — 304 с.
- *Макконнелл С.* Совершенный код. Мастер-класс. — М.: Издательство «Русская редакция», 2010. — 896 стр.
- *ДеМарко Т.* Deadline. Роман об управлении проектами. — М.: Манн, Иванов и Фербер. — 2013. — 352 с.

Дополнительная литература:

- *Соммервилл И.* Инженерия программного обеспечения. — Издательство Вильямс, 2002. — 624 с.
- *ДеМарко Т., Листер Т.* Человеческий фактор. Успешные проекты и команды. — М.: Символ-Плюс. — 2014. — 288 с.

Ключевые стандарты, руководства, своды знаний

- SWEBOOK: Guide to the Software Engineering Body of Knowledge
<http://www.computer.org/web/swebok>
- ISO/IEC/IEEE 12207:2017 System and software engineering — Software life cycle processes
- CMMI for Development (CMMI-DEV), Version 1.3
<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9661>
- Essence — Kernel and Language for Software Engineering Methods
<http://www.omg.org/spec/Essence>
- ISO/IEC/IEEE 16326:2019 Systems and software engineering — Life cycle processes — Project management
- SWECOM: IEEE Software Engineering Competency Model
<https://www.computer.org/volunteering/boards-and-committees/professional-educational-activities/software-engineering-competency-model>
- ISO/IEC/IEEE 24765:2017 Systems and software engineering — Vocabulary
https://pascal.computer.org/sev_display/index.action

Виды обеспечения вычислительной системы

- **Математическое:** методы, модели, алгоритмы.
- **Лингвистическое:** языки, способы взаимодействия.
- **Информационное:** данные.
- **Программное:** программы.
- **Техническое:** аппаратные средства.
- **Методическое:** пользовательская документация и т.д.
- **Организационное:** правила, регламенты, процедуры.
- **Правовое:** правовые нормы

(ГОСТ 34.003-90 Информационная технология.

Автоматизированные системы. Термины и определения)

Программа —

- **исполняемая программа:** комбинация инструкций и данных, позволяющая аппаратному обеспечению компьютера выполнять вычисления или функции управления (*ISO/IEC/IEEE 24765:2010 Systems and software engineering — Vocabulary*).
- **исходный текст:** синтаксическая единица, которая соответствует правилам определённого языка программирования (*ISO/IEC 2382-1:1993, Information technology — Vocabulary — Part 1: Fundamental terms*).

Программное обеспечение, программный продукт

Программное обеспéчение — программа или множество программ, используемых для управления компьютером (*IEEE Std 829—2008*).

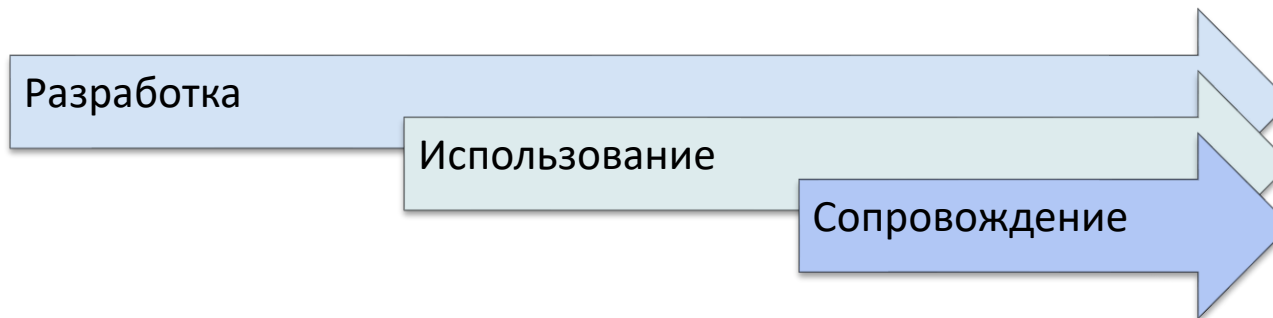
Программный продукт — законченная программная система, готовая для продажи (поставки) потребителю вместе с процедурами, документацией и данными.

Complete set of software designed for delivery to a software consumer or end-user which can include computer programs, procedures and associated documentation and data (*ISO/IEC 19770-5:2015*)

Для создания системного программного продукта требуется примерно **в девять раз больше усилий** по сравнению с составляющими его программами, написанными отдельно для частного использования (*Ф. Брукс*).

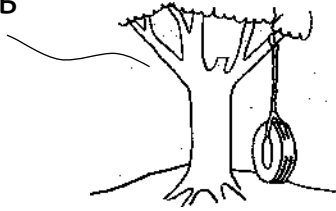
Программная инженерия (Software engineering)

Программная инженерия — применение систематического, упорядоченного, измеримого подхода к разработке, использованию и сопровождению программного обеспечения, то есть приложение дисциплины инженерии к программному обеспечению (*ISO/IEC/IEEE 24765*)

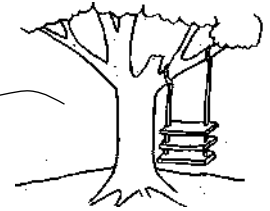


Кризис ПО

1. Чего хотел пользователь



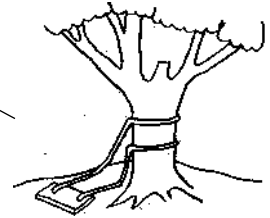
2. Как пользователь сумел объяснить, чего он хочет



3. Как было описано в техническом задании



4. Как было спроектировано



5. Как было реализовано программистами



6. Как было внедрено



7. Как прорекламовал отдел маркетинга



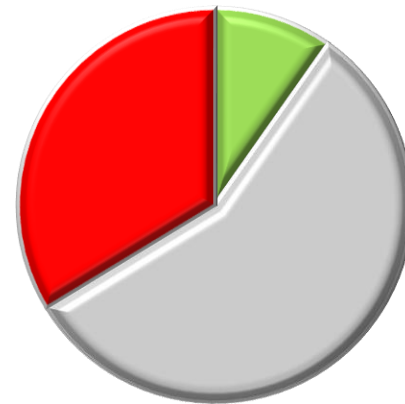
8. Как было оплачено



Кризис ПО

В 1995 г. признаны успешными 9% проектов;
52% проектов превысили бюджет на 189 %;
31% проектов провалились (статистика по
крупным проектам).

Исследования The Standish Group



Проектов:	2004	2006	2008	2010	2012	2015
Успешных, %	29	35	32	37	39	29
Проблемных, %	18	19	24	21	18	52
Провалов, %	53	46	44	42	43	19

См. «выдающиеся» примеры (*List of failed and overbudget custom software projects*):

https://en.wikipedia.org/wiki/List_of_failed_and_overbudget_custom_software_projects

Сделать то, что нужно и сделать хорошо

Продукт...	Нужный	Ненужный
Хорошо сконструированный	Успех	Бизнес-провал
Плохо сконструированный	Кошмар сопровождения	Бесполезная дрянь

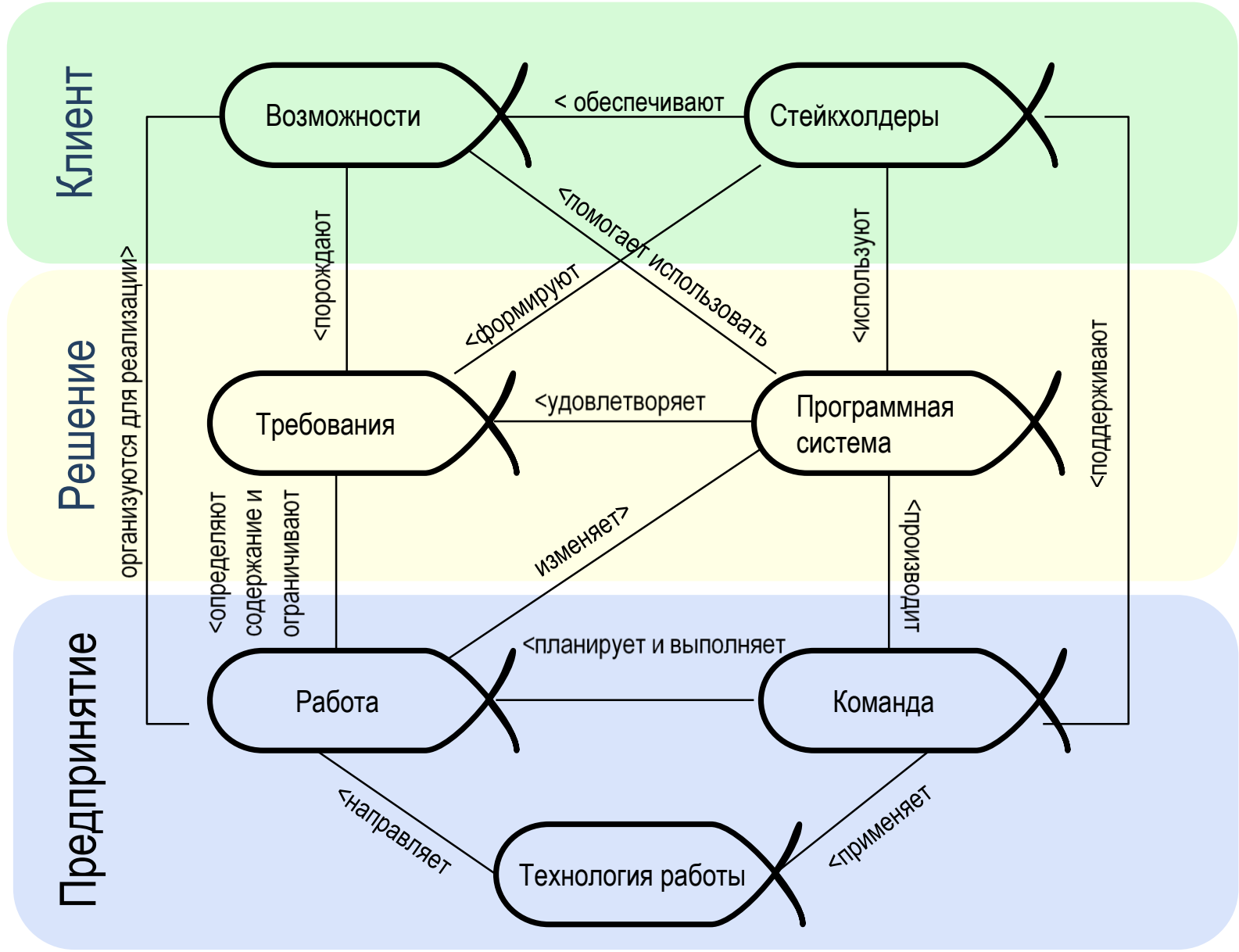
Факторы успеха и провала проекта

Фактор успеха	Ранг, %
1. Вовлечение стейкхолдеров	15,9
2. Поддержка со стороны руководства	13,9
3. Чётко определённые требования	13
4. Правильное планирование	9,6
5. Реалистичные ожидания	8,2
6. Деление на маленькие этапы	7,7
7. Компетентные сотрудники	7,2

Фактор провала	Ранг, %
1. Неполные требования	13,1
2. Плохое вовлечение стейкхолдеров	12,4
3. Недостаток ресурсов	10,6
4. Нереалистичные ожидания	9,9
5. Слабая поддержка со стороны руководства	9,3
6. Изменение требований и спецификаций	8,7
7. Плохое планирование	8,1

The Chaos Report, 2014,
The Standish Group International, Inc.

Альфы инженерного проекта (стандарт OMG Essence)



Сложность разработки ПО



Ключевая статья — “No Silver Bullet” Фредерика Брукса, 1987 г.

Неустранимые причины сложности разработки ПО (примеры):

1. Сложность предметной области.
2. Внутренняя сложность программ.
3. Отсутствие хороших способов представления больших систем.
4. Трудности управления процессом разработки.
5. Изменение требований к программе в процессе её разработки.

Жизненный цикл программного средства

Жизненный цикл (ЖЦ) программного средства — процесс последовательного изменения его состояния от момента возникновения необходимости в нём до полного вывода из эксплуатации.

Процесс ЖЦ — систематически выполняющийся и целенаправленный вид деятельности.

Стадия (этап) ЖЦ — период в пределах жизненного цикла программного средства, который характеризует его состояние и ключевую инженерную деятельность.

Модель ЖЦ — определённое деление ЖЦ на **процессы** и **стадии (этапы) ЖЦ**, которые могут перекрываться или повторяться циклически.

Жизненный цикл программного продукта

Ключевые стандарты:

- ISO/IEC 15288:2015 Systems and software engineering — System life cycle processes
- ISO/IEC/IEEE 12207:2017 System and software engineering — Software life cycle processes (ГОСТ Р ИСО/МЭК 12207—2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств)

Типовые модели жизненного цикла (примеры)

Стандарт ISO/IEC 15288:2002 (см. русскоязычный аналог ГОСТ Р ИСО МЭК 15288-2005):

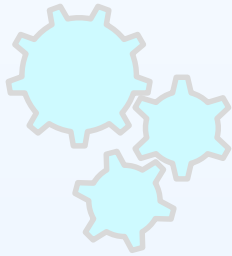
1. Стадия замысла.
2. Стадия разработки.
3. Стадия производства.
4. Стадия применения.
5. Стадия поддержки применения.
6. Стадия прекращения применения и списания.

Rational Unified Process:

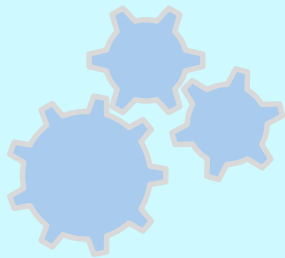
1. Начало (Inception)
2. Проработка (Elaboration)
3. Конструирование (Construction)
4. Передача (Transition)

Группы процессов жизненного цикла (ISO 15288)

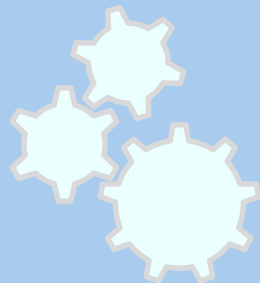
Процессы организации



Процессы проекта



Технические процессы



Процессы соглашения



- Agreement Processes
- Organizational Project-Enabling Processes
- Technical Management Processes
- Technical processes