

Введение в программную инженерию

Наумов Д.А., доц. каф. КТ

Основы программной инженерии, 2019

Содержание лекции

- 1 История и основные понятия
- 2 Отличия программной инженерии от других отраслей
- 3 Эволюция подходов к управлению

Программная инженерия есть применение определенного систематического измеримого подхода при разработке, эксплуатации и поддержке программного обеспечения.

- *software* (программное обеспечение, ПО) - 1958, Джон Тьюкей (John Tukey)
- *software engineering* (программная инженерия) - конференция НАТО, Германия, 1968
- с 1990 по 1995 год: работа над международным стандартом о процессах разработки ПО (ISO/IEC 12207)
- "Руководство к своду знаний по программной инженерии" (SWEBOK), 2004

Программирование - процесс отображения определенного множества целей на множество машинных команд и данных, интерпретация которых на компьютере или вычислительном комплексе обеспечивает достижение поставленных целей.

- цели → требования к системе
- требования → высокоуровневая архитектура и спецификации компонентов
- спецификации → на дизайн компонентов
- дизайн → на исходный код
- исходный код → код развертывания
- код развертывания → вызовы функций ПО окружения (ОС, промежуточное ПО, базы данных), которое может располагаться на множестве компьютеров, объединенных в сеть
- вызовы функций ПО окружения → машинные команды и данные.

Профессиональное программирование

деятельность, направленная на получение доходов при помощи программирования.

Профессиональное производство программ это всегда коллективная деятельность, в которой участвуют минимум два человека: программист и потребитель.

Профессиональный программист

человек, который занимается профессиональным программированием.

Жизненный цикл программного продукта



Процесс разработки ПО

совокупность процессов, обеспечивающих создание и развитие программного обеспечения.

Модель процесса разработки ПО

формализованное представление процесса разработки ПО.

Часто при описании процессов вместо слова *модель* употребляется термин *методология*, что приводит к неоправданному расширению данного понятия.

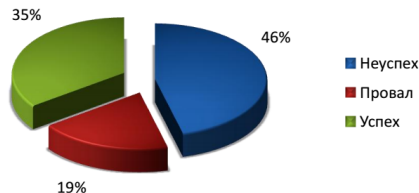
Основные области знаний программной инженерии (согласно SWEBOOK 2004)

- 1 Software requirements – программные требования.
- 2 Software design – дизайн (архитектура).
- 3 Software construction – конструирование программного обеспечения.
- 4 Software testing – тестирование.
- 5 Software maintenance – эксплуатация (поддержка) программного обеспечения.
- 6 Software configuration management – конфигурационное управление.
- 7 Software engineering management – управление в программной инженерии.
- 8 Software engineering process – процессы программной инженерии.
- 9 Software engineering tools and methods – инструменты и методы.
- 10 Software quality – качество программного обеспечения.

Дополнительные области знаний программной инженерии (согласно SWEBOOK 2004)

- 1 Computer engineering – разработка компьютеров.
- 2 Computer science – информатика.
- 3 Management – общий менеджмент.
- 4 Mathematics – математика.
- 5 Project management – управление проектами.
- 6 Quality management – управление качеством.
- 7 Systems engineering – системное проектирование.

Результаты анализа успешности программных проектов за 2006 год



- Только 35% проектов завершились в срок, не превысили запланированный бюджет и реализовали все требуемые функции и возможности.
- 46% проектов завершились с опозданием, расходы превысили запланированный бюджет, требуемые функции не были реализованы в полном объеме.
- 19% проектов полностью провалились и были аннулированы до завершения.

Кто виноват? Что делать?

Ф. Брукс, 1975:

Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки шлифовки или переработки и доступен для воплощения грандиозных замыслов.

- программирование - наука?
- программирование - искусство?
- программирование - ремесло?
- программирование - сходство и отличия с другими областями знаний

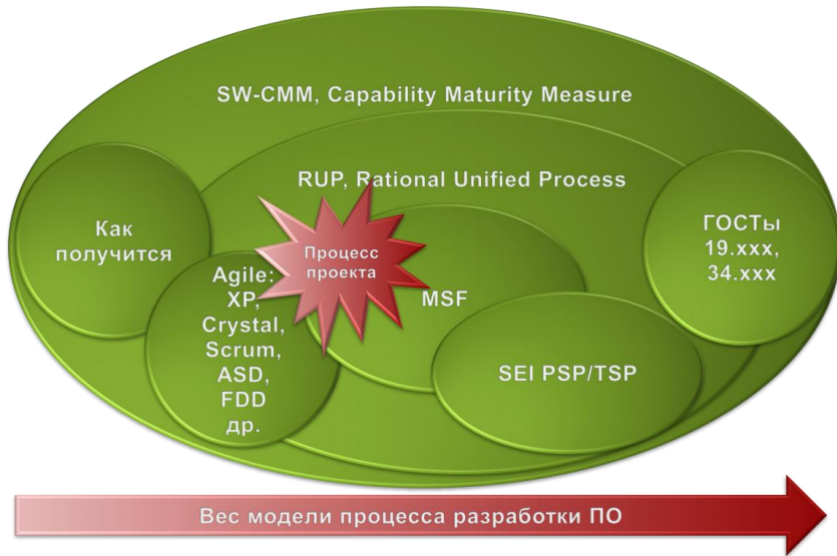
- *Как получится.* Полное доверие техническим лидерам. Представители бизнеса практически не участвует в проекте. Планирование, если оно и есть, то неформальное и словесное. Время и бюджет, как правило, не контролируются.
- *Водопад* или каскадная модель. Жесткое управление с обратной связью.
- *Гибкое управление.* Гибкое управление с обратной связью
- *Метод частых поставок.*

Классические методы управления перестают работать, когда:

- структура и свойства управляемого объекта нам не известны и/или изменяются со временем
- рабочая группа проекта не может обеспечить требуемую эффективность

Когда структура и свойства управляемого объекта нам не известны, необходимо использовать адаптивное управление, которое, дополнительно к прямым управляющим воздействиям, направлено на изучение и изменение свойств управляемого объекта.

Различные модели процесса разработки ПО и их распределение по "весу"



ГОСТы (ГОСТ 19 "Единая система программной документации" и ГОСТ 34 "Стандарты на разработку и сопровождение автоматизированных систем")

- ориентированы на последовательный подход к разработке ПО
- разработка проводится по этапам, каждый из которых предполагает выполнение строго определенных работ, и завершается выпуском достаточно большого числа весьма формализованных и обширных документов
- строгое следование не только приводит к водопадному подходу, но и требует очень высокой степени формализованности разработки.
- на основе этих стандартов разрабатываются программные системы по госзаказам в России.

SW-CMM (Capability Maturity Model for Software)

- данная модель определяет пять уровней зрелости процесса разработки ПО (Начальный, Повторяемый, Определенный, Управляемый, Оптимизируемый).
- документация с полным описанием занимает около 500 страниц и определяет набор из 312 требований, которым должна соответствовать организация, если она планирует аттестоваться по этому стандарту на 5-ый уровень зрелости.

RUP (Rational Unified Process)

- разработан в качестве дополнения к языку моделирования UML
- описывает абстрактный общий процесс, на основе которого организация или проектная команда должна создать конкретный специализированный процесс, ориентированный на ее потребности
- можно использовать и как основу для самого что ни на есть традиционного водопадного стиля разработки, так и в качестве гибкого процесса

Microsoft Solutions Framework (MSF)

- это гибкая и достаточно легковесная модель, построенная на основе итеративной разработки
- большое внимание к созданию эффективной и небюрократизированной проектной команды
- предлагает достаточно нестандартные подходы к организационной структуре, распределению ответственности и принципам взаимодействия внутри команды

PSP/TSP (Personal Software Process / Team Software Process)

- делает ставку на самоуправляемые команды численностью 3–20 разработчиков
- определяет требования к компетенциям команд в целом и программистов

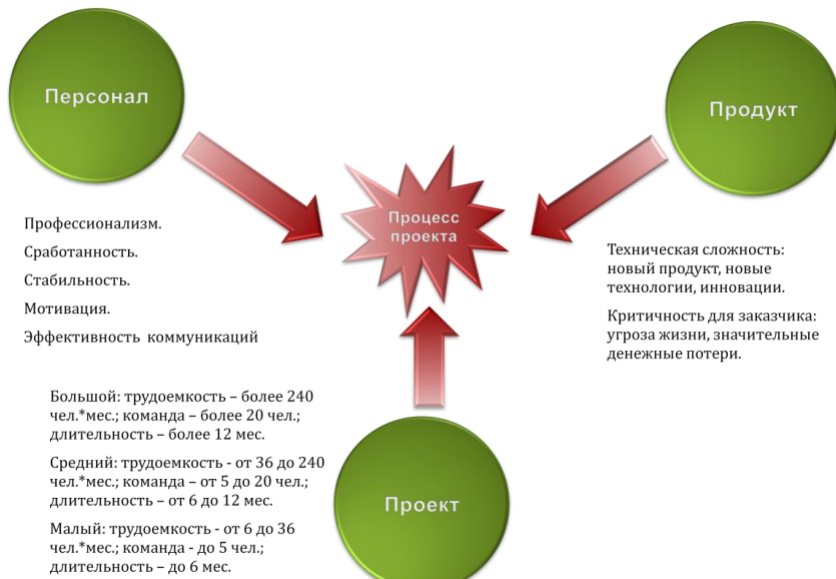
Каждый программист должен уметь:

- учитывать время, затраченное на работу над проектом;
- учитывать найденные дефекты и классифицировать типы дефектов; оценивать размер задачи;
- осуществлять систематический подход к описанию результатов тестирования;
- планировать программные задачи; распределять их по времени и составлять график работы.
- выполнять индивидуальную проверку проекта и архитектуры;
- выполнять регрессионное тестирование.

Agile (гибкая разработка ПО)

- применяемый в разработке ПО процесс должен быть адаптивным
- высшей ценностью является ориентированность на людей и их взаимодействие, а не на процессы и средства
- гибкие методологии это - набор практик, которые могут позволить (а могут и нет) добиваться эффективной разработки ПО, основываясь на итеративности, инкрементальности, самоуправляемости команды и адаптивности процесса.

Процесс в проекте должен определяться в зависимости от проекта, продукта и персонала



Что надо делать для успеха программного проекта? (Стив Макконнелл, «Остаться в живых. Руководство для менеджеров программных проектов», «Питер», 2006)

- 1 Четко ставить цели.
- 2 Определять способ достижения целей.
- 3 Контролировать и управлять реализацией.
- 4 Анализировать угрозы и противодействовать им.
- 5 Создавать команду.

Выводы

- То, что производят программисты нематериально – это коллективные мысли и идеи, выраженные на языке программирования. В силу уникальности отрасли опыт, накопленный в отраслях материального производства, мало способствует успеху в управлении программным проектом. Прямые аналогии с этими отраслями не работают. Управлять разработкой ПО надо иначе.
- Не существует единственного правильного процесса разработки ПО. Эффективный производственный процесс должен основываться на итеративности, инкрементальности, самоуправляемости команды и адаптивности. Главный принцип: не люди должны строиться под выбранную модель процесса, а модель процесса должна подстраиваться под конкретную команду, чтобы обеспечить ее наивысшую производительность.

- ① IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
- ② IEEE Std 1074-1995, IEEE Standard for Developing Software Life Cycle Processes.
- ③ "Руководство к своду знаний по программной инженерии". The Guide to the Software Engineering Body of Knowledge, SWEBOK, IEEE Computer Society Professional Practices Committee, 2004.
- ④ David Rubinstein, "Standish Group Report: There's Less Development Chaos Today". 2007
(<http://www.sdtimes.com/content/article.aspx?ArticleID=30247>)
- ⑤ Брукс Фредерик, "Мифический человеко-месяц, или Как создаются программные комплексы Пер. с англ., СПб., Символ-Плюс, 1999
- ⑥ "PMBOK. Руководство к Своду знаний по управлению проектами 3-е изд., PMI, 2004.
- ⑦ Уолкер Ройс, "Адаптивный стиль управления программными проектами". Открытые системы. 2006. № 1.

- 8 Ершов А. П., "О человеческом и эстетическом факторе в программировании". Информатика и образование. 1993. № 6.
- 9 Paulk, Mark C., and others, Capability Maturity Model for Software, Version 1.1 (CMU/SEI-93-TR-24). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University, 1993.
- 10 Филипп Крачтен, "Введение в Rational Unified Process Вильямс, 2002 г.
- 11 "MSF, Microsoft, Microsoft Solutions Framework Отдел MSF, Microsoft, 2002.
- 12 M. Pomeroy-Huff, J. Mullaney, R. Cannon, M. Sebern, "The Personal Software Process (PSP) Body of Knowledge version 1.0, SPECIAL REPORT CMU/SEI, 2005
- 13 Watts S. Humphrey, "The Team Software Process (TSP) Technical Report CMU/SEI, 2000
- 14 Kent Beck, and others, "Manifesto for Agile Software Development 2001 (<http://www.agilemanifesto.org/>)

- 15 А. Коуберн, "Люди как нелинейные и наиболее важные компоненты в создании программного обеспечения Humans and Technology Technical Report, Oct.1999
- 16 А. Коуберн, "Каждому проекту своя методология Humans and Technology Technical Report, TR 99.04, Oct.1999 (русский перевод К.Максимов, А.Максимова).
- 17 С. Макконнелл, "Остаться в живых. Руководство для менеджеров программных проектов "Питер 2006.