

Принципи в практиката

■ Принципи, които ръководят практиката

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

Лектор: Доц. д-р Ася Стоянова-Дойчева

Software Engineering Knowledge

- *Често сте чувате хората да казват, че знанията за разработка на софтуер имат 3 години живот: половината от това, което трябва да знаете днес ще бъде остаряло след 3 години. Относно знанията за технологиите, това най-вероятно е истина. Но има друг вид знания в софтуерната разработка – вид знания, които наричам "**software engineering principles**"— които нямат 3 години живот. Тези принципи на софтуерната разработка могат да служат на един професионален програмист през цялата му кариера*

Steve McConnell

Принципи, които управляват процеса- I

- **Принцип #1. Бъди гъвкав.** Без значение какъв модел на процеса сте избрали основните принципи на гъвкавата разработка следва да ръководят подхода.
- **Принцип #2. Фокусиране върху качеството на всяка стъпка.** Крайната цел за всяка дейност, действие, задача в процеса е да се фокусира върху качеството на крайния продукт.
- **Принцип #3. Бъдете готови да адаптирате.** Процеса не е религиозен опит или догма, която трябва да спазваме. Когато е необходимо трябва да адаптираме подхода към ограниченията възникнали от проблема, хората и от самия проект.
- **Принцип #4. Създайте ефективен екип.** Процеса и практиката са важни, но в основата на успеха са хората. Изградете самоорганизиращ се екип, който има взаимно доверие и уважение.

Принципи, които управляват процеса- II

- **Принцип #5. Създаване на механизми за комуникация и координация.** Проектите пропадат, защото важна информация се губи и заинтересованите страни не координират техните изисквания за създаване на успешен краен продукт.
- **Принцип #6. Управление на промените.** Подходът може да бъде формален или не формален, но трябва да бъде създаден механизъм за начина на управление на промените, оценени, одобрени и приложени.
- **Принцип #7. Оценка на риска.** Много неща могат да се объркат по време на създаване на софтуер. Затова трябва да създадете план как да работите в извънредно положение
- **Принцип #8. Създавайте работещи продукти, които дават стойност на други.** Създаване само тези работни продукти, които осигуряват стойност за други процесни дейности, дейности или задачи.

Принципи, които ръководят практиката

- **Принцип #1. Разделяй и владей.** От техническа гледна точка анализа и проектирането трябва да се правят на отделни части (*separation of concerns* (SoC)).
- **Принцип #2. Разбиране на използването на абстракция.** Това е основата, абстракцията е опростяване на някакъв сложен елемент на системата. В анализа и проектирането се използват абстракции на високо ниво.
- **Принцип #3. Стреми се към съгласуваност.** Познатия контекст прави софтуера по-лесен за използване.
- **Принцип #4. Съсредоточаване върху трансфера на информация.** Отделя се специално внимание на анализа, проектирането, разработката и тестването на интерфейса.

Принципи, които ръководят практиката

- **Принцип #5. Изграждане на софтуера на ефективни модули.** Разделянето на части (Принцип #1) установява философията за софтуера. Модулирането предоставя механизма за реализация на философията.
- **Принцип #6. Търсене на шаблони.** Brad Appleton [App00] предлага: “Целта на шаблоните е да се създаде основа, помагаща на софтуерните инженери да решават повтарящи се проблеми...”
- **Принцип #7. Когато е възможно проблемът и неговите решения да се представят от различни перспективи.**
- **Принцип #8. Помнете, че някой ще поддържа софтуера.**

Комуникационни принципи

- **Принцип #1. *Слушай.*** Опитай да се фокусираш върху думите на клиента, отколкото върху формулиране на твоите отговорности спрямо казаното.
- **Принцип # 2. *Подготви се преди да комуникираш.*** Прекарай време за разбиране на проблема преди да се среща.
- **Принцип # 3. *Някой трябва да управлява дейностите.*** Винаги срещата трябва да има лидер (a facilitator), който да направлява срещата и разговорите в продуктивна посока; (2) да е медиатор при всеки възникнал конфликт, и (3) да осигури, че останалите принципи се спазват.
- **Принцип #4 *Комуникация лице в лице е най-добра.*** Тази комуникация работи добре, ако има и някакво друго представяне на информацията.

Комуникационни принципи

- **Принцип# 5. Прави записки и документи.** Някой от участниците в комуникацията трябва да записва всички важни точки и решения.
- **Принцип# 6. Стреми се към сътрудничество.** Сътрудничество и консенсус се постигат, когато знанията на членовете на колектива се обединят и комбинират.
- **Принцип# 7. Остани съсредоточен.** Много хора в дискусията прескачат от тема на тема.
- **Принцип# 8. Ако нещо не е ясно, нарисуй го ☺**
- **Принцип# 9. (a) Ако си се съгласил с нещо, продължи напред; (b) Ако не можеш да се съгласиш с нещо, продължи напред; (c) Ако характеристика или функция не е изяснена и не може да бъде изчистена в момента, продължи напред.**
- **Принцип# 10. Преговорите не са състезание или игра. Те са най-добри, когато и двете страни печелят.**

Принципи на планирането

- **Принцип #1. *Разбиране на обхвата на проекта.*** Невъзможно е да направиш или използваш карата, ако не знаеш къде отиваш. Обхвата и целта на проекта дават посока на екипа.
- **Принцип #2. *Включване на клиента в планирането.*** Клиентът дефинира приоритетите и определя ограниченията в проекта.
- **Принцип #3. *Планирането е итеративно.*** Плана на проекта не е константа. При започване на работата е много вероятно той да се промени.
- **Принцип #4. *Приблизителна оценка на базата на това, което знаете.*** Целта на оценяването е да се предоставят индикатори за усилията, цената, и продължителността на задачите, на базата на текущите знания на екипа за работата, която трябва да бъде **свършена.**

Принципи на планирането

- **Принцип #5. Пресметнете риска след като дефинирате плана.** Ако идентифицирате рискове с голямо влияние и голяма вероятност, е необходимо планиране за извънредни ситуации.
- **Принцип #6. Бъдете реалисти.** Хората не работят на 100% всеки ден.
- **Принцип #7. Регулиране на детайлността при дефиниране на плана.** Детайлността се отнася за нивото на детайлите, които са записани в проектния план за реализация.
- **Принцип #8. Дефиниране на това как ще се гарантира качеството.** Планът трябва да определя как софтуерния екип ще гарантира качеството.
- **Принцип #9. Описание как възнамеряваме да се справим с промените.** Дори и най-доброто планиране може да бъде провалено от неконтролирани промени.
- **Принцип #10. Следене на плана и внасяне на промени според изискванията.**

Принципи на моделирането

- В софтуерното инженерство могат да бъдат създадени два класа модели:
 - *Моделите на изискванията* (наричани още *аналитични модели*), които представят изискванията на клиента като ги определя в три различни домейна: информационен домейн, функционален домейн и домейн на поведението.
 - *Проектните модели* представят характеристиките на софтуера които са необходими за реализацията: архитектура, потребителски интерфейс, и детайли на компонентно ниво.

Принципи на моделите на изисквания

- Принцип #1. *Информационния домейн на проблема трябва да бъде представен и разбран.*
- Принцип #2. *Функциите, които софтуерът изпълнява трябва да бъдат дефинирани.*
- Принцип #3. *Поведението на софтуера (като следствия от външни събития) трябва да бъде представено.*
- Принцип #4. *Моделите, които представят информация, функционалност и поведение, трябва да бъдат разпределени върху малки части от проблема, които да могат да бъдат разбрани.*
- Принцип #5. *Аналитичните задачи трябва да водят от обща информация към детайли на реализацията.*

Принципи на проектното моделиране

- Принцип #1. Проекта трябва да бъде проследим в модела на изискванията.
- Принцип #2. Винаги трябва да се има предвид, че архитектурата на системата ще се разработва.
- Принцип #3. Проектирането на данните е толкова важно, колкото и функциите за обработка.
- Принцип #4. Интерфейсите (вътрешни и външни) трябва да бъдат проектирани внимателно.
- Принцип #5. Проектирането на потребителският интерфейс трябва да бъде направено според нуждите на потребителя. Във всеки случай той трябва да е удобен и лесен за използване.
- Принцип #6. Проекта на компонентното ниво трябва да бъде функционално независим – силна съгласуваност.
- Принцип #7. Компонентите трябва да бъдат слабо свързани един с друг и към външната среда.
- Принцип #8. Проектното представяне (моделите) трябва да бъде лесно разбираемо.
- Принцип #9. Проектирането трябва да се развива итеративно. С всяка итерация проектантът трябва да се стреми към по-голяма простота.

Agile принципи за моделиране

- Принцип #1. Основната цел на екипа е да създаде софтуер, а не да създава модели.
- Принцип#2. Не се създават повече модели, от колкото са необходими.
- Принцип#3. Стремех към създаване на най-простите модели, които ще опишат проблема или софтуера.
- Принцип#4. *Изграждане на модели по начин, който ги прави податливи на промените.*
- Принцип#5. *Посочване на целта на всеки модел, който е създаден.*
- Принцип#6. *Адаптиране на моделите.*
- Принцип#7. Опитайте се да създавате полезни модели, но не перфектни модели.
- Принцип#8. Не ставайте догматични относно синтаксиса на моделите, *ако те вършат работа.*
- Принцип#9. Ако инстинктите ви казват, че модела не е правилен, въпреки че изглежда добре, най-вероятно имате причина да се безпокоите.
- Принцип#10. Обратна връзка.

Принципи за разработката

- Дейностите на разработката включват кодиране и задачи на тестване, които водят до създаване на софтуер готов за доставка на клиента или крайният потребител.
- **Понятията и принципите на кодирането** са тясно обвързани - стил на програмиране, език за програмиране и методи за програмиране.
- **Понятията и принципите на тестването** водят до проектиране на тестове, които систематично откриват различни класове грешки, като го правят за минимално време и усилия.

Принципи на подготовка

■ ***Преди да напишете и един ред код бъдете сигурни:***

- Разбирате проблема, който се опитвате да решите.
- Разбирате основните принципи и понятия на проектирането.
- Избор на език за програмиране, който отговаря на нуждите на софтуера, който се разработва и на средата, в която ще работи.
- Избор на среда за програмиране, която предоставя средства, които ще направят работата ви по-лесна.
- Създаване на множество от unit тестове, когато компонента, който пишете е завършен.

Принципи на кодирането

- **Когато започнете да пишете кода бъдете сигурни:**
 - Обмисляне използването на pair programming
 - Избиране на структури от данни, които отговарят на изискванията на проекта.
 - Разбиране на софтуерната архитектура и създаване на интерфейси, които са съвместими с нея.
 - Запазете условната логика колкото е възможно по-проста.
 - Създавайте вложени цикли в случаи че са лесни за тестване.
 - Избирайте смислени имена на променливи и следвайте други локални стандарти за кодиране.
 - Пишете код, който се самодокументира
 - Добра четаемост.

Принципи за валидация

- ***След като сте завършили първия си пас на кодиране, бъдете сигурни:***
 - Пробвайте кода, когато е удобно.
 - Проведете unit тестове и поправете грешките, които откриете.
 - Направете рефакторинг на кода.

Принципи за тестване

- Al Davis [Dav95] предлага следното:
 - **Принцип #1.** Всички тестове трябва да бъдат проследими до клиентските изисквания.
 - **Принцип #2.** Тестовете трябва да бъдат планирани много преди тестването да започне.
 - **Принцип #3.** *Принципът на Pareto важи и при тестването.*
 - **Принцип #4.** Тестването трябва да започне “в малкото” и продължава към тестване “в голямото”.
 - **Принцип #5.** *Изчерпателно тестване не е възможно.*

Deployment Принципи

- **Принцип #1. Очакванията на клиента трябва да бъдат управлявани.** Много често клиентът очаква повече от това което екипа е обещал да достави и има разочарование.
- **Принцип #2. Готовият продукт трябва да бъде сглобен и тестван.**
- **Принцип #3. *Трябва да бъде установен режим за поддръжка преди да бъде доставен.*** Крайният потребител очаква отзивчивост и точна информация при възникване на въпрос или проблем.
- **Принцип #4. Подходящи материали с инструкции трябва да бъдат предоставени на крайните потребители.**
- **Принцип #5. Софтуер с бъгове първо трябва да се поправи и след това да се предостави на клиента.**