

Въведение в софтуерните технологии.

Лекция 1

Доц. д-р Ася Стоянова-Дойчева



Съдържание

- Какво е софтуерен продукт
- Характеристики на софтуера
- Възникване на дисциплината Софтуерни технологии
- Какво е софтуерен процес
- Модели на софтуерния процес
 - Линейни последователни
 - Непоследователни
- Литература



Какво е софтуерен продукт

- Какво е софтуерен продукт?

????????????

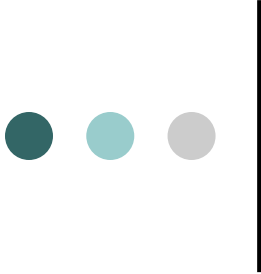
Софтуерен продукт =

програми + данни + документация



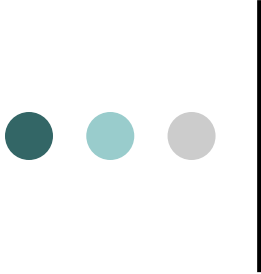
Ролята на софтуера

- Софтуерът има двойнствена роля
 - Продукт – доставя компютърен потенциал. Той е информационен преобразовател.
 - Средство за доставяне на продукт – действа като основа за контрол на компютъра(ОС), трансфер на информация (мрежи) и за създаване на друг софтуер (софтуерни средства и среди)



Характеристики на софтуера

- *Софтуера се разработва (engineered), той не се произвежда в класическия смисъл.*
- *Абстрактност*
- *Мултидисциплинираност*
- *Надеждност*
- *Рискове*
- *Софтуера не се “износва”.*
- *Компонентно базирано разработване*



Възникване на дисциплината Софтуерни технологии

- Software Engineering – НАТО 1969г.
 - Софтуерната криза
- Еквивалентен термин на български е Софтуерни технологии



Дефиниции

- “Software Engineering е дисциплина, която предоставя методи и средства за разработване на качествен софтуер с ограничен бюджет, поставени срокове и с постоянно изменящи се изисквания. “
- “*Software engineering* - приложение на систематичен, дисциплиниран и определен подход за разработването, изпълнението и поддръжката на качествен софтуер.”

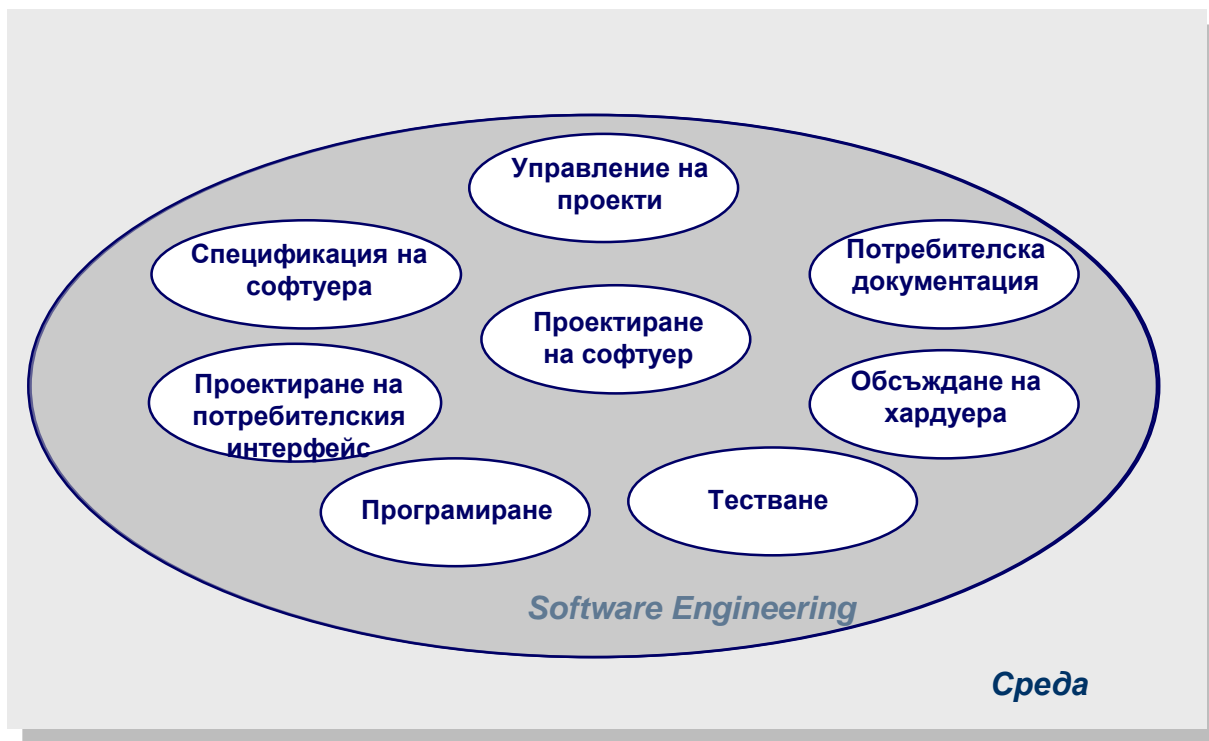
IEEE



Цели на дисциплината

- Лесна поддръжка
- Надеждност на софтуера
- Ефективност на софтуера
- Удобен и лесен за използване
- Цена

Области включени в софтуерните технологии

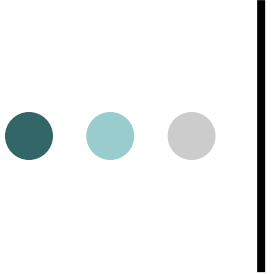




Области включени в предмета на софтуерните

ТЕХНОЛОГИИ

- life cycle models
- software processes (CMM, ISO 9000)
- configuration management
- effort estimation
- project management
- quality management
- cleanroom software engineering
- reverse engineering
- program comprehension (program understanding)
- software metrics
- software architecture
- software maintenance
- requirements engineering
- software specification
- reuse
- program verification
- software documentation
- knowledge-based SE
- systematic testing
- prototyping
- CASE tools
- programming languages for SE
- object-orientation (analysis, design, programming, test)
- extreme programming
- pair programming



Защо дисциплината софтуерни технологии е важна ????

- ???????



Запитване до младите специалисти в компютърните науки, работещи в практиката от Германската Асоциация по Информатика

Важност на подобластите
на информатиката. (най-
важните са първи в
списъка)

subareas of informatics	score
team work	3.59
project management	3.35
software engineering	3.26
leadership	3.10
databases	3.10
rhetorics	3.09
communication systems / computer nets	2.99
quality assurance	2.96
data structures and efficient algorithms	2.81
operating systems	2.60
software ergonomics	2.56
business management	2.50
algorithmic fundamentals / complexity	2.42
mathematical and logical fundamentals	2.41
distributed systems	2.36
scientific work methods	2.36
data security	2.32
information systems	2.31
concepts of programming languages	2.27
computer architecture	1.87
legal fundamentals	1.68
disposition systems	1.64
analysis and assessment of computer systems	1.64
graphics and visualization systems	1.58
modeling and simulation	1.53
formal languages and automaton	1.37
real-time systems	1.30
multimedia	1.27
informatics and society	1.15
fundamentals of electronics	1.13
machine-level programming	1.06
CIM	1.04
expert systems	0.88
fundamentals of artificial intelligence	0.85
electrical measurement	0.81
image processing	0.81
applications of artificial intelligence	0.81
pattern recognition	0.63

Задачите на софтуерните технологии през призмата на офертите за работа



CSC PLOENZKE

Projektmanager

Als Projektmanager sind Sie verantwortlich für die termin-, kosten- und qualitätsgerechte Abwicklung von großen Lösungsprojekten für unsere Kunden aus Industrie und Verwaltung.

Die wirtschaftliche Herbeiführung der Kundenzufriedenheit ist Ihr oberstes Ziel. Hierzu motivieren Sie Ihr Team, beziehen die Unterstützungsleistungen von CSC Europe und gegebenenfalls CSC USA mit ein und steuern Ihr Projekt wie ein Unternehmen auf Zeit.

Ръководител на проект

Като ръководител на проект вие сте отговорен за реализацията на сложни проекти за нашите клиенти в индустрията и управлението. Главната задача е да **удовлетворите изискванията на клиента, като същевременно проекта е реализиран икономично...**

Задачите на софтуерните технологии през призмата на офертите за работа

*Wir sind ein auf Bankensoftware spezialisiertes Softwarehaus im Großraum München mit namhaften Kunden im In- und Ausland. Unser Produkt **OBS Online Banken System** unterstützt alle Bereiche des internationalen Bankgeschäfts, besonders im Handels- und Wertpapierbereich.*

Zur Verstärkung unseres jungen Teams suchen wir

APPLIKATIONSENTWICKLER/IN

Ihre Aufgaben:

- Analyse der fachlichen Anforderungen
- Design der Anwendungen
- Programmierung
- Test und Qualitätssicherung
- Implementation beim Kunden

Ihr Profil:

- Cobolkenntnisse oder vergleichbare Kenntnisse
- Kenntnisse eines der folgenden Systeme:
UNIX, Open VMS, OS/400
- SQL und Datenbankkenntnisse, z. B. Oracle
- Kaufmännische Grundkenntnisse

Задачи:

- анализ на техническите изисквания
- проектиране на приложения
- програмиране
- тестване и осигуряване на качеството
- инсталация на зададената среда

Вашият профил:

- Опит с COBOL
- Умения в:
UNIX, Open VMS, OS/400
- SQL и бази данни - Oracle
- търговски умения

Задачите на софтуерните технологии през призмата на офертите за работа

Sie verfügen über eine mehrjährige Erfahrung in Software-Entwicklungsprojekten. Darüber hinaus erwarten wir fundierte Programmiererfahrungen in C, OpenVMS, UNIX und Networking.

Außerdem suchen wir eine/n

Software-Entwickler/in

Ihr Tätigkeitsbereich umfaßt:

- Analyse bestehender Software-Funktionalität
- Machbarkeitsuntersuchungen und Aufwandsschätzungen
- Mitarbeit an technischen Spezifikationen
- Erweiterung existierender Software-Module

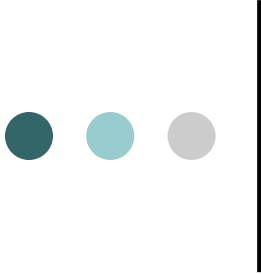
Разработчик на софтуер

Задачи:

- анализ на изискванията на дадена система
- прилагане на анализ и оценка на необходимите усилия
- техническа спецификация
- разширение на съществуващи софтуерни модули
- регресивно тестване и обновяване на софтуерни процедури

hrungen in C, DCL und
nwendungen. Erfahrun-

Schrift. Darüber hinaus

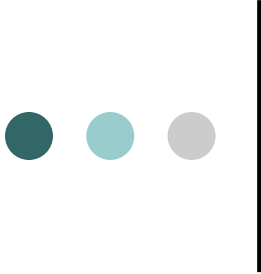


Защо е важна дисциплината софтуерни технологии в практиката

Софтуерът е предразположен към грешки:

- Работи по друг начин различен от очаквания
- Обикновено е по-скъп от планираното
- Разработката му приключва твърде късно
- Не е приложим

Много често софтуерните проекти пропадат!



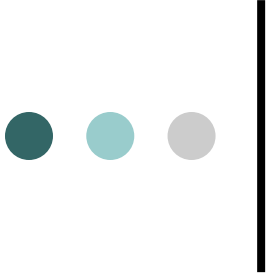
Примери за софтуерни проблеми

Летището в Денвър – не е отворило през 1994 – софтуера за транспортиране на багажа не е работил.

Deutsche Telekom – грешка в изчисленията на таксата за разговори на 01.01.96 (софтуерна грешка: използвана е тарифата за почивен ден - загуби: 100 милиона DM)

Siemens - забавена е разработката на счетоводен софтуер за медикаменти разработван за здравни осигуровки, което довело до допълнителни разходи от 1 билион DM - Berliner Zeitung 22.05.96

Първата космическа ракета до Венера – през 1979 тя не достига целта: в програма на Фортран те използвали точка вместо запетая. Загубите възлизат на няколко стотици милиона долара. (Wallmüller 1990, p. 1).



Какво е софтуерен процес – жизнен цикъл на програмния продукт

- Жизнен цикъл на програмния продукт – целият период на неговото създаване и използване.
 - Начало – възникване на идеята за създаване на продукта
 - Край – моментът в който се преустановява използването му.



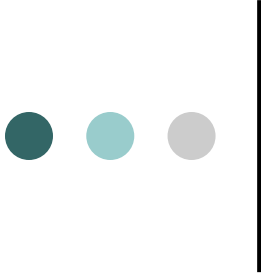
Софтуерен процес

Дефиниция: Софтуерен развоен процес
(software development process) -
цялостният процес на поставяне на
задачата, планиране, реализиране и
оценка на едно софтуерно и хардуерно
приложение, включително и
използваните помощни средства,
методи и необходимия персонал.



Софтуерен процес

- Фаза на дефиницията – задава се въпроса Какво?
 - Оценка на информацията
 - Планиране на софтуерния проект
 - Анализ на изискванията
- Фаза на разработването – задава се въпроса Как?
 - Софтуерно проектиране
 - Генериране на кода
 - Тестване на софтуера
- Фаза на поддръжката.

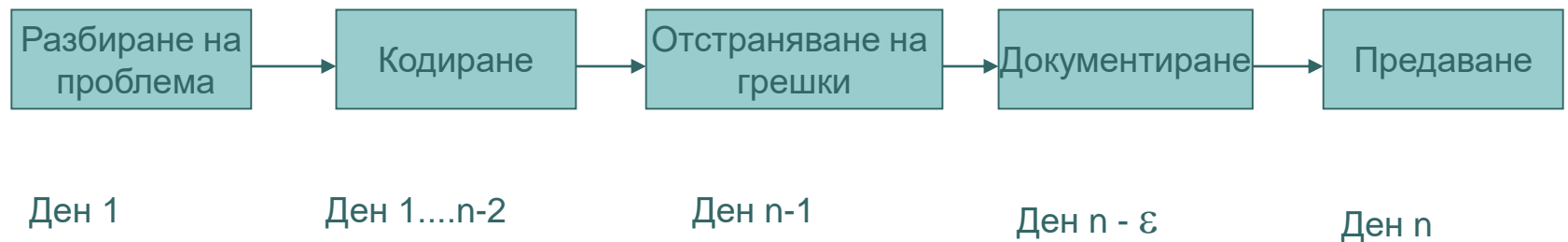


Разлики на развойния процес от ad-hoc програмирането

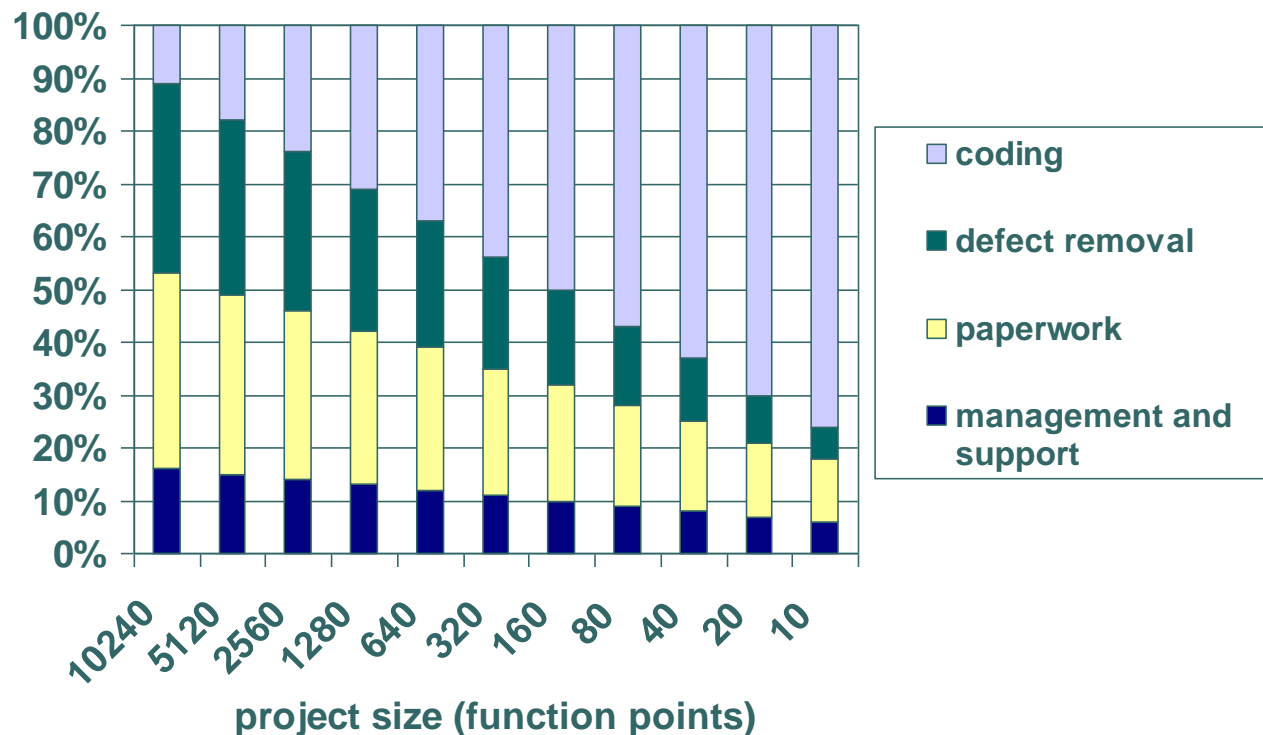
- Изисква се системен анализ
- Участват повече хора
- Софтуерът се разработва обикновено за външни потребители
- Развойният процес се извършва поетапно (жизнен цикъл на процеса)

Основа за реализиране на един такъв процес са развойните методи (software development methods)

Как студентите виждат разработката на софтуер



Как действително протича разработката на софтуер





Модели на софтуерния процес

Стратегия за разработване, която включва процес, методи, средства и основните фази за разработка се нарича модел на софтуерния процес.

Избора на модел на СП става на базата на вида на приложението, методите и средствата, които ще бъдат използвани и изискванията към приложението.



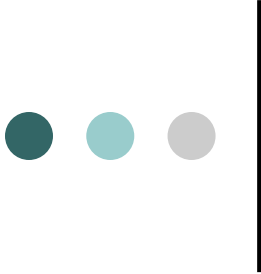
Защо са нужни модели

- Правят процеса методологичен
- Спомагат за определянето на организацията на екипа
- Дават технологични насоки



Модели на софтуерния процес

- **Линейни последователни**
- Непоследователни модели
- Гъвкави модели (Agile Software Process).

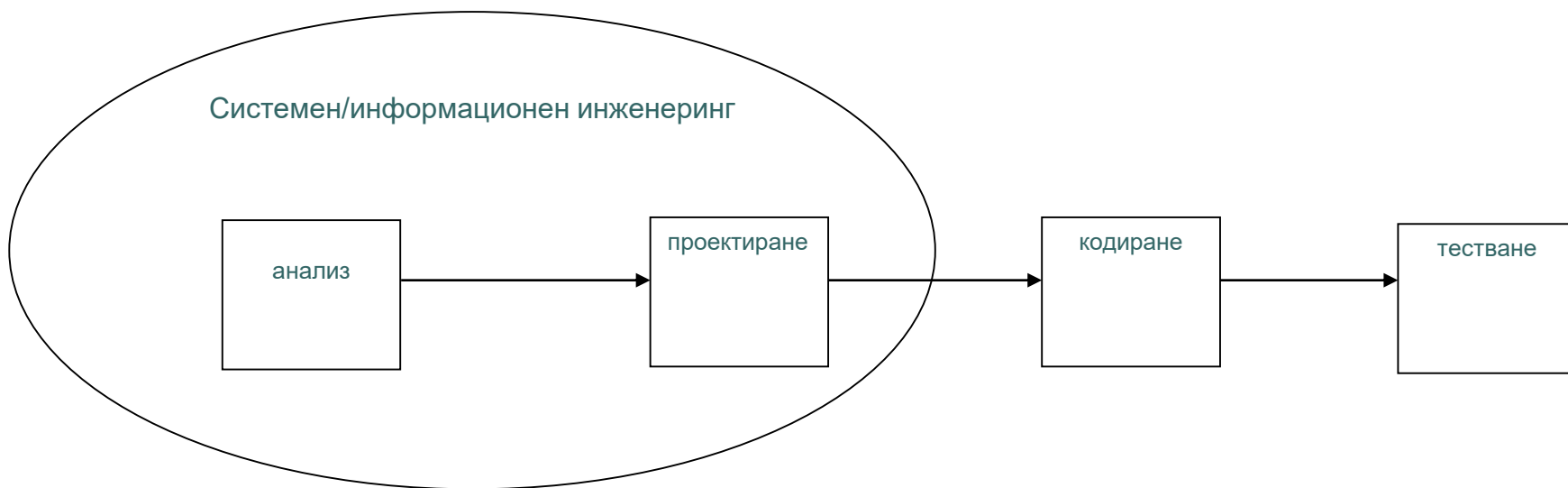


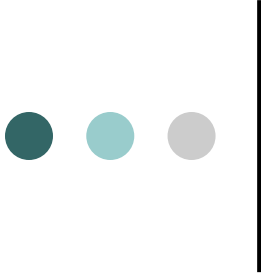
Линейни последователни модели

- Водопаден модел (класически жизнен цикъл) - линейните последователни модели предлагат систематичен, последователен подход за разработване на софтуер, който започва от системното ниво и прогресира през анализ, проектиране, кодиране, тестване и поддръжка.

Водопаден модел

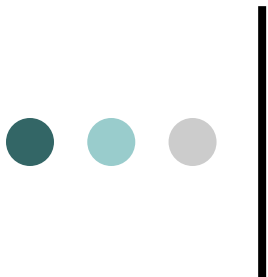
- Включва следните дейности:



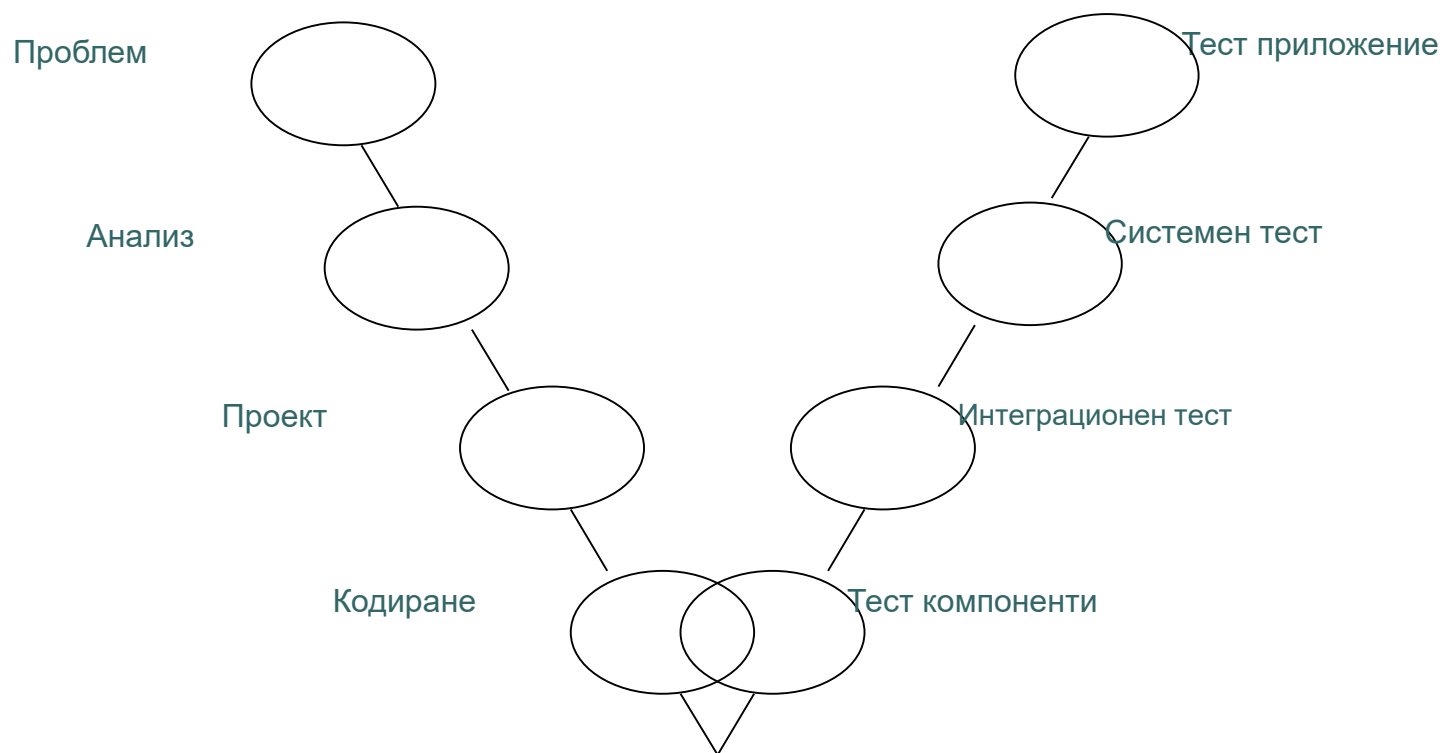


Недостатъци на водопадния модел

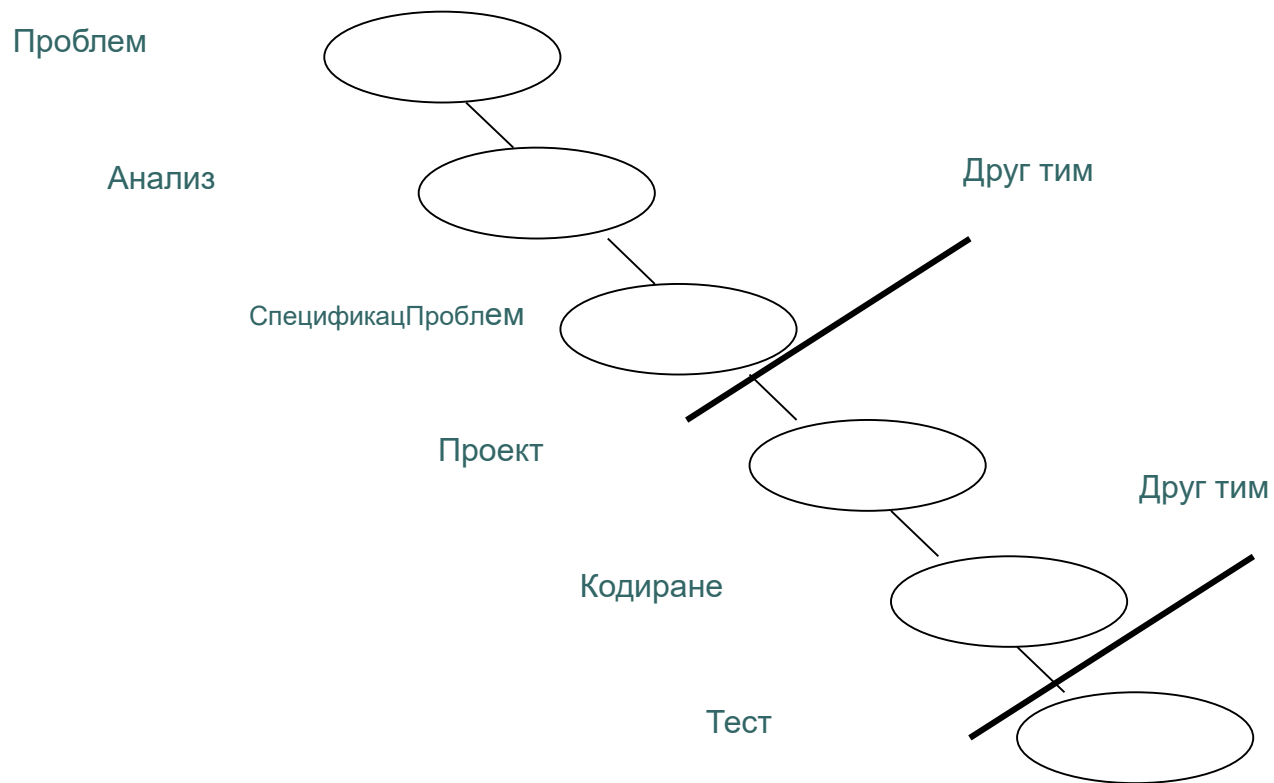
- Реалните проекти рядко следват последователния поток на модела
- Често е трудно за потребителите да поставят своите изисквания за софтуера наведнъж – което е изискване за този тип модели
- Потребителите трябва да са търпеливи – работеща версия на системата ще бъде налична не по-рано от крайния срок за изпълнение на проекта.



V-модел



Cleanroom-engineering



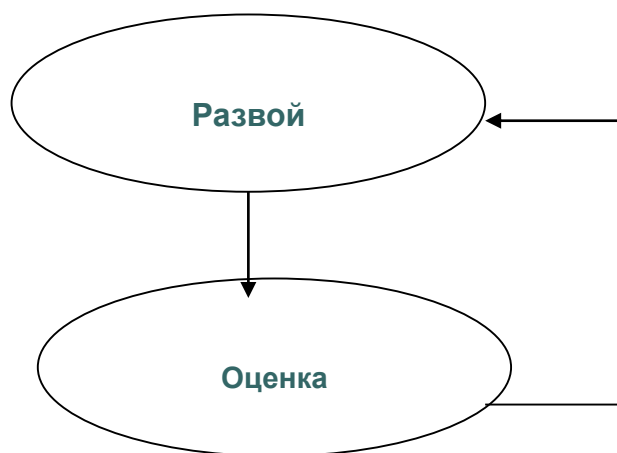


Модели на софтуерния процес

- Линейни последователни
- Непоследователни модели
- Гъвкави модели (Agile Software Process)

Непоследователни модели

- Еволюционен развой



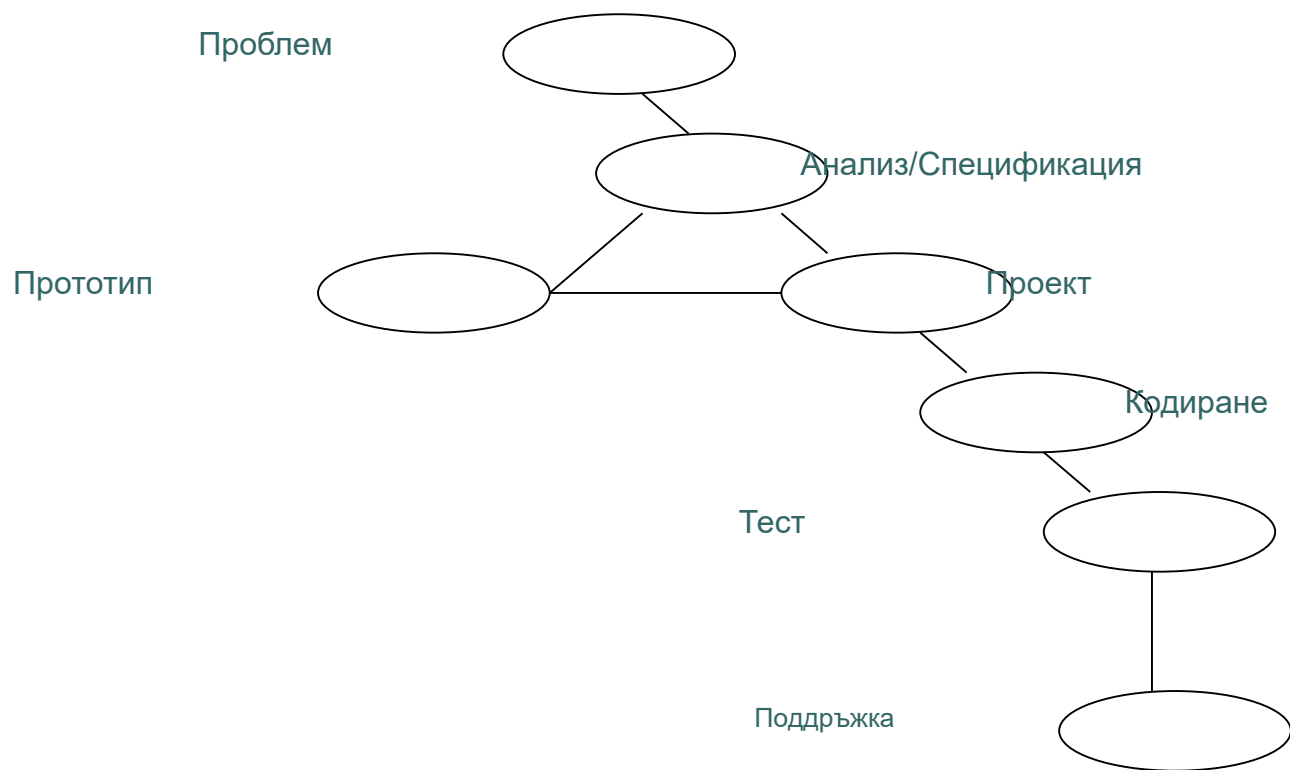


Прототипиране

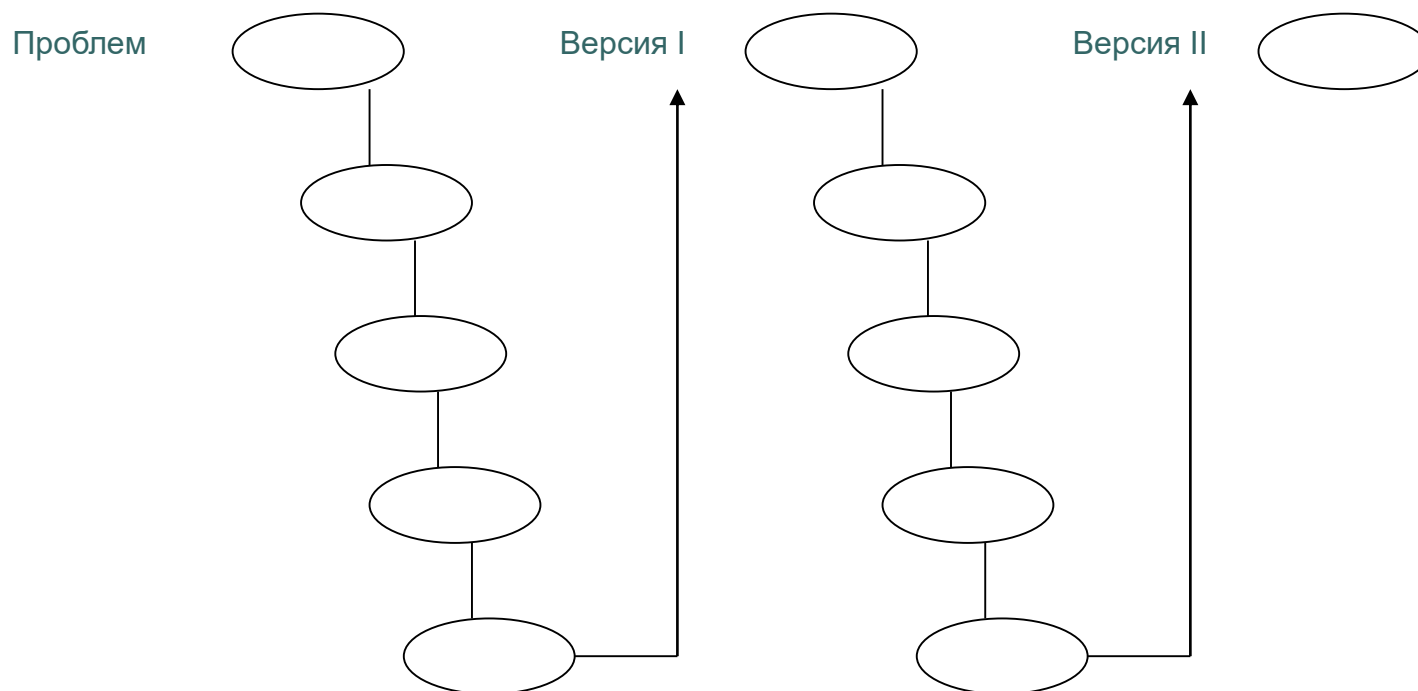
Специална форма на еволюцията е прототипирането.

- Вертикално прототипиране - избират се само някои функции, които обаче се реализират напълно.
- Хоризонтално прототипиране - разглежда се цялата функционалност, реализира се обаче само в общи щрихи.

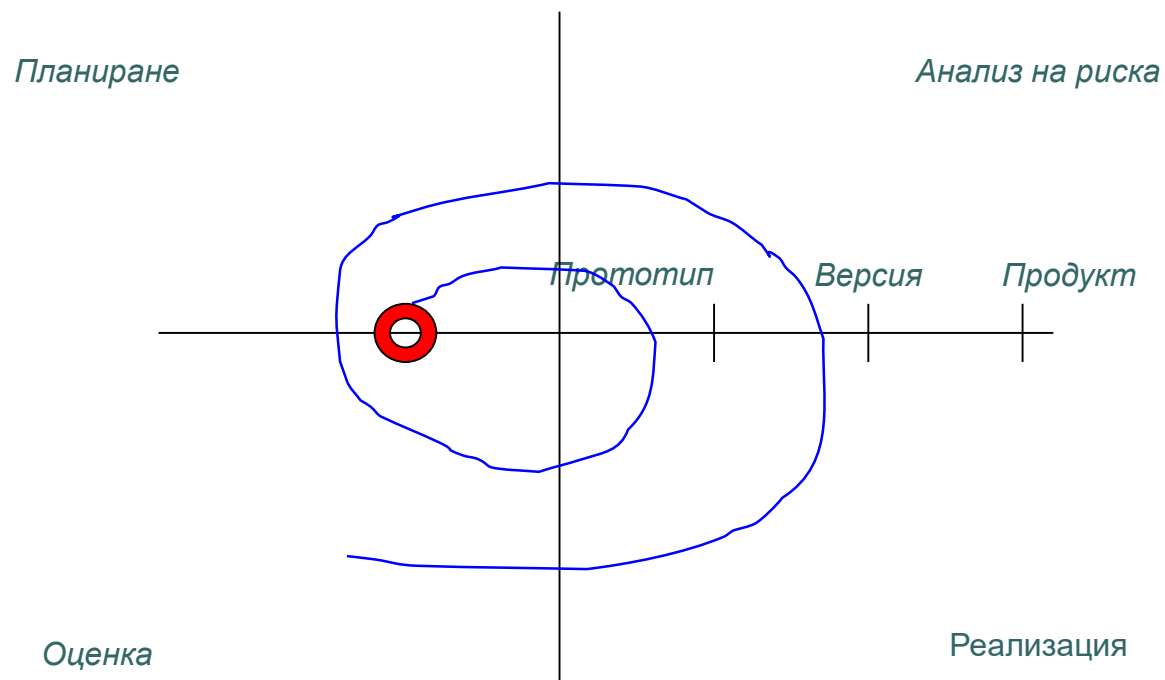
Прототипиране



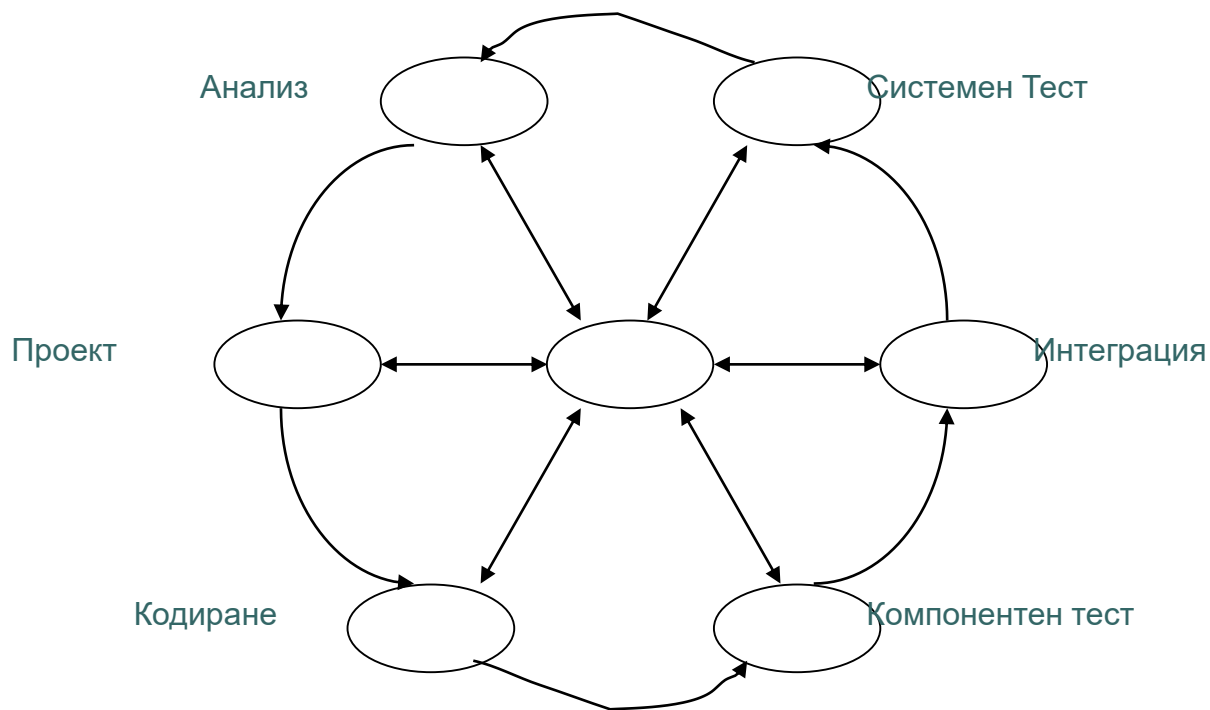
Инкрементален развой



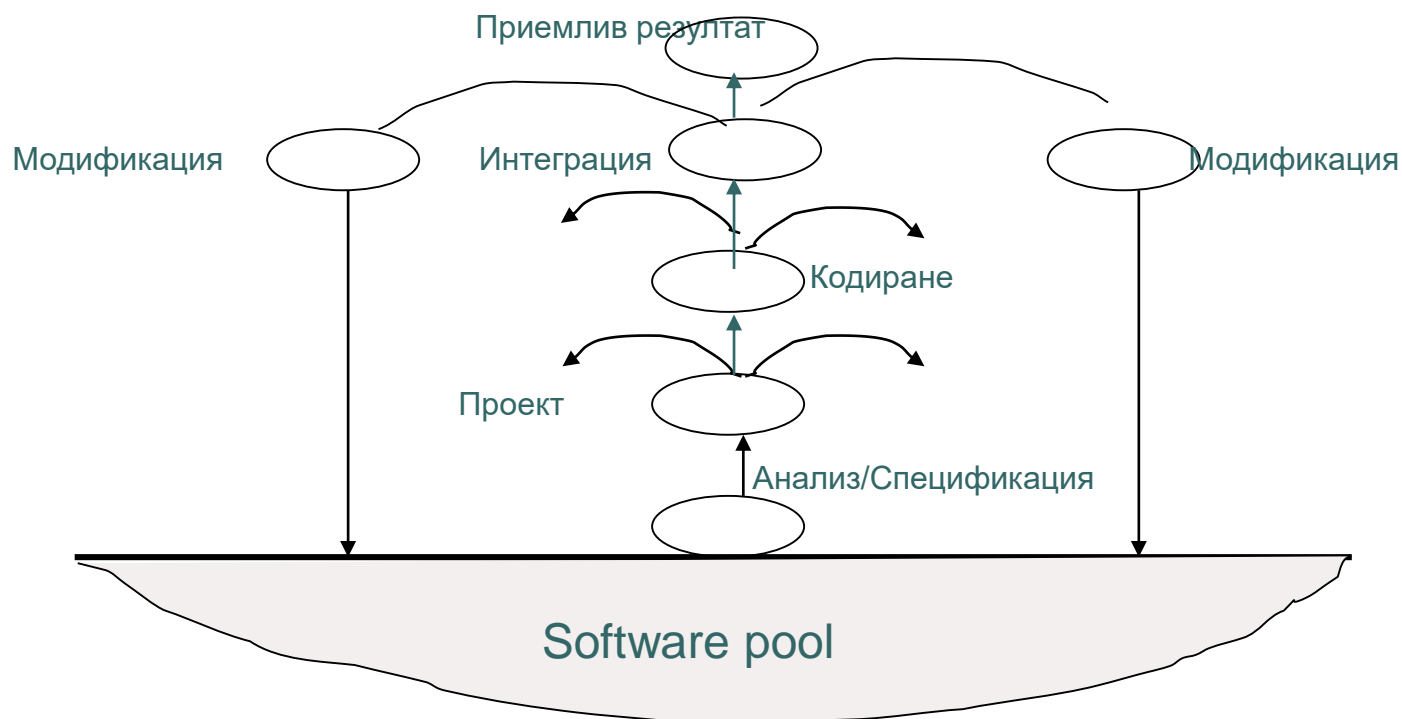
Спирален модел



Whirlpool модел



Фонтанен модел





Модели на софтуерния процес

- Линейни последователни
- Непоследователни модели
- Гъвкави модели (Agile Software Process).

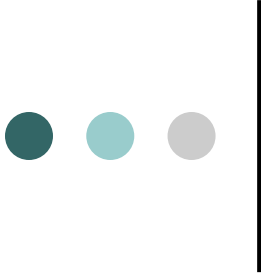
Какво е Agile software process?



agile - прилагателно

пъргав	snappy, agile, nimble, versatile, spry, lively
подвижен	movable, mobile, moveable, moving, flexible, agile
бърз	hurried, rush, rapid, fast, quick, agile
ловък	agile, adroit, slick, skillful, shrewd, able
сръчен	skillful, skilful, handy, dexterous, deft, agile
жив	alive, living, live, lively, vivid, agile

- **Agile** в софтуерната индустрия – гъвкав, откликващ на промените, интензивен
- *Гъвкав подход за разработка на софтуер*



Характеристики на Agile Software Process

- Трудно е да се предвиди кои софтуерни изисквания ще са постоянни и кои ще се променят;
- За много типове софтуерни продукти проектирането и разработката се препокриват. Това са две дейности, които се изпълняват заедно и проектните модели се доказват, че работят, докато се разработват;
- Анализа, проектирането, разработката и тестването не са толкова предсказуеми, колкото би ни се искало.



Agile Software Process

- Възниква въпроса как да управляваме непредсказуем процес???
- Чрез адаптивност на процеса! – Но постоянна адаптивност без прогрес се постига трудно. Следователно Agile SP трябва да се *адаптира инкрементално*.
 - *Обратна връзка с клиента;*
 - *Инкрементална стратегия за разработка;*
 - *Доставяне на малки работещи прототипи (software increments) на малки интервали от време до клиента.*



Принципи при Agile

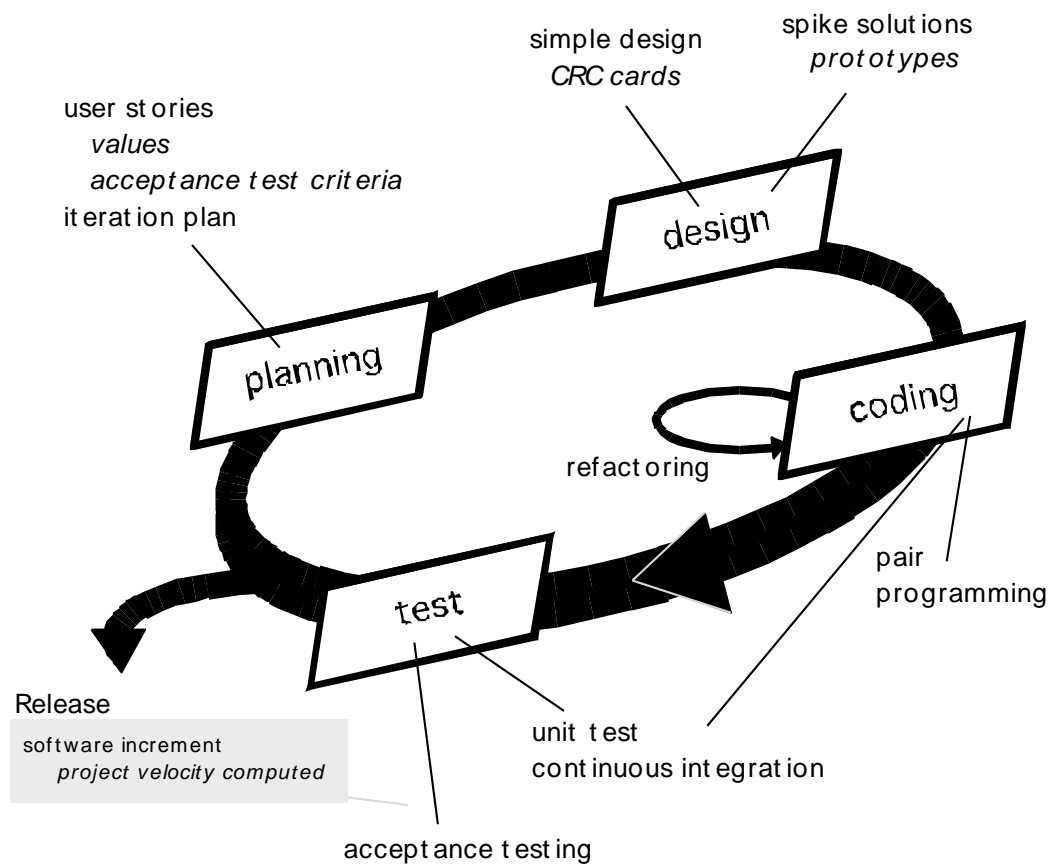
1. Най-важният приоритет е да се удовлетворят изискванията на клиента;
2. Промените се правят дори късно в процеса на разработка на продукта;
3. Доставят се работещи версии на продукта на няколко седмици или месец;
4. Клиентите и разработчиците работят заедно по време на целия проект;
5. Проекта се разработва от мотивирани хора, на които се осигуряват необходимите средства и им се вярва, че ще свършат работата си;
6. Най-ефективния начин за разпространение и обмяна на информация между членовете на екипа е директния разговор;



Принципи на Agile

7. Работещия софтуер е основна мярка за прогрес;
8. Agile процесите насърчават устойчивата разработка – толерира се постоянно темпо;
9. Набляга се на качествена разработка – техническа и проектна;
10. Простота – максимално увеличаване на обема на работа – толкова, колкото може да се свърши;
11. Най-добрите архитектура, изисквания и проект възникват в самоорганизиращ се екип;
12. На определени интервали екипа отразява как да стане по-ефективен.

Extreme Programming



SCRUM

SCRUM





Заключение

Софтуерния инженеринг е дисциплина, която интегрира процеси, методи и средства за разработване на софтуер. Различни модели на процеса за разработка на софтуер са предложени като всеки има своите преимущества и недостатъци, но всеки от тях има серия от базови стъпки, които ще бъдат разгледани в рамките на този курс.



Литература

- Textbooks:

- Нели Манева, Аврам Ескенази, Софтуерни технологии. Анубис София 2001
- I. Sommerville: Software Engineering (742 pages), 6. ed. Addison-Wesley, 2001
- H. Balzert: Lehrbuch der Software-Technik, Spektrum-Verlag, Vol. 1: 1996, Vol. 2: 1998 (769 pages), Vol. 1 (2nd ed.): 2001 (1136 pages)
- B.-U. Pagel, H.-W. Six: Software Engineering. Addison-Wesley (895 pages), 1994 (in German)
- E. Braude: Software Engineering, An Object-Oriented Perspective, Wiley & Sons, 2001
- R. Pressman: Software Engineering, A Practitioner's Approach, European Adaptation (840 pages), McGraw-Hill, 5th ed., 2000
- J. Marciniak: Encyclopedia of Software Engineering, Vol. 1, 2 (1453 pages), Wiley & Sons, 1994
- IEEE Standards Collection, Software Engineering, 1994 Edition



Литература

- Списания:

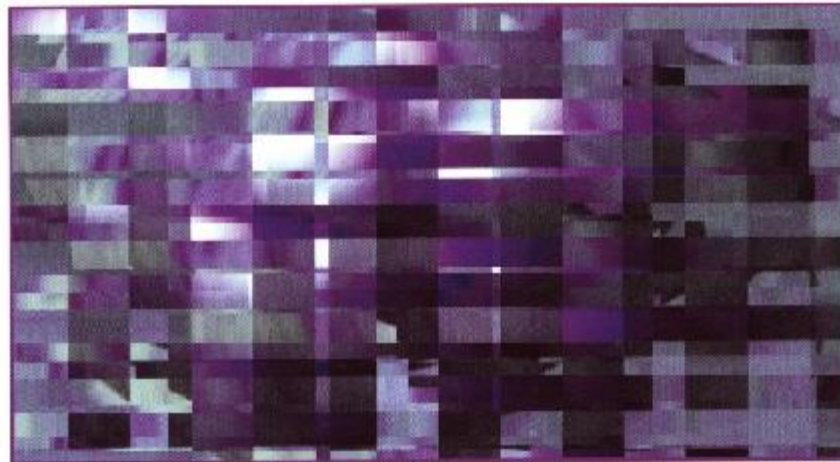
- IEEE Transactions on Software Engineering
- IEEE Software
- ACM Software Engineering Notes
- ACM Transactions on Software Engineering and Methodology

- Конференции:

- International Conference on Software Engineering
- Conference on Software Engineering Education
- Software Metrics Symposium
- and many others

Software Engineering

A Practitioner's Approach
European Adaptation



Fifth Edition

Roger S. Pressman

adapted by
Darrel Ince

Нели Манева
Аврам Ескенази

СОФТУЕРНИ ТЕХНОЛОГИИ





Software Engineering

A PRACTITIONER'S APPROACH

EIGHTH EDITION

Roger S. Pressman, Ph.D.
Bruce R. Maxim, Ph.D.

Mc
Graw
Hill
Education