

Жизнен цикъл



на информационна система



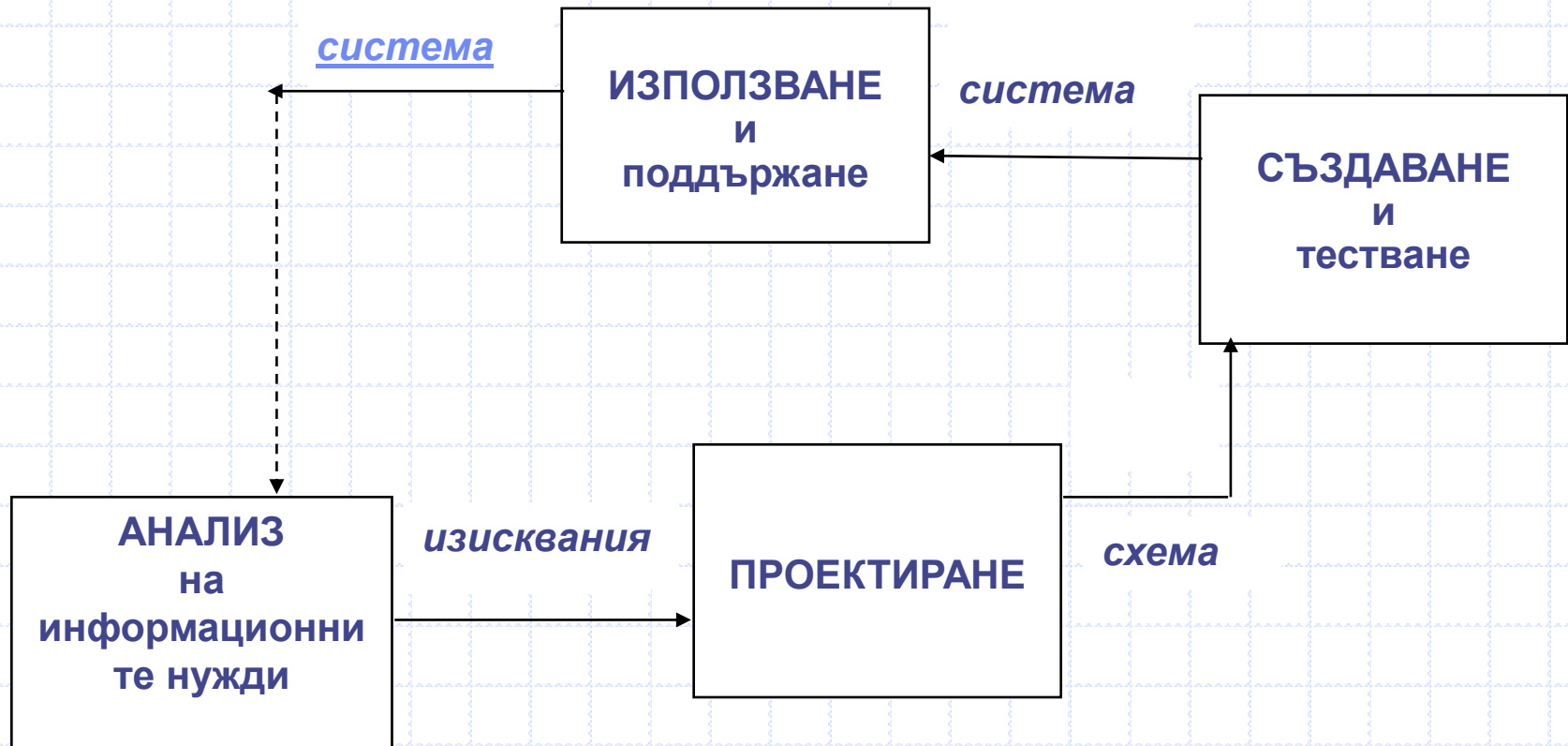
Жизнен цикъл

- **Определение**
 - Формален процес на създаване на информационна система, следващ определени правила, процедури и стандарти
- **Съдържание**
 - състои се от етапи (фази)
 - всеки етап има
 - определена цел
 - средства и
 - изпълнители

Жизнен цикъл

- Всеки етап се състои от стъпки, които
 - имат определен обект и цел
 - завършват с конкретен резултат
 - прилагат подходящи технологии
 - се реализират от определени тип участници
 - се документират

Етапи на жизнения цикъл



Анализ

- **Цел:** дефиниране на изисквания
- **Обект:** потребителите и техните информационни нужди
 - “крайни” потребители: ползватели
 - “системни” потребители: програмисти, администратори
- Информационни нужди -> изисквания към системата
 - ползвателите задават **съдържанието**
 - професионалистите определят **структурата**

Анализ

- На езика на бизнеса
- Въпроси
 - защо е необходима информационна система, каква ще бъде нейната стойност за бизнеса?
 - кой и как ще я използва?
 - къде и кога ще бъде използвана?
 - кой и как ще я създаде – план, персонал, техника, управление

Анализ

- Основни стъпки
 - стратегия за анализ
 - събиране на данни
 - дефиниране на изисквания
 - инициране на проект

Анализ

- Основни дейности

- Събиране на данни – събиране на документи, проучване, наблюдение, интервю
- Анализ на фактите – статистика, приоритети, сложност, трудоемкост, риск
- Установяване на недостатъци на съществуващата система
- Изследване за осъществимост
 - техническа
 - финансова
 - организационна
 - етична, от социална гледна точка
- Инициране на проект

Анализ

- **Резултати**
 - Дефиниране на изисквания – БД, софтуер, хардуер
 - Избор на алтернативен модел
 - Изготвяне на прототип, който се оценява от потребителите
- **Средства**
 - Current environment logical data structure (ERD)
 - Data Flow Diagram DFD
 - flowcharts, connectivity diagram, grid, decision table, prototypes

Проектиране

- Цел
 - Изготвяне на проект
 - Установяване на среда за функциониране и контрол
- Дейности
 - Стратегия на проектирането
 - Избор на системна архитектура – хардуер, софтуер, мрежи
 - логическо проектиране
 - физическо проектиране
- Резултати
 - Проект на базите от данни
 - Проект на управляващите програми
 - Проект на потребителските интерфейси
 - Проект на средата за функциониране
- Средства
 - CASE (Computer Aided Software Engineering)

Проектиране

- Въпроси
 - как ще оперира системата?
 - в каква среда ще оперира
 - хардуер
 - софтуер
 - мрежи
 - какъв потребителски интерфейс е необходим
 - екранни форми
 - печатни документи
 - какви специфични програми и данни са необходими?

Създаване

- Цел
 - реализиране на системата
- Дейности
 - Програмиране
 - Тестване
 - Сертифициране
 - + Инсталиране на хардуер
 - Обучение на потребителите

Тестване

- План на тестване
- Обекти на тестване
 - съдържанието
 - как се въвеждат данните, правила и ограничения
 - структурата
 - съдържа ли БД структурата от модела на данни
 - съдържа ли физическата структура (индекси, таблици на символите, правила за достъп и др.)
 - поведението
 - тестване на процедури и програмни модули
- Тестови данни
 - специално подбрани данни
 - реални данни

Сертифициране

- Същност

- определяне на използваемостта в различни приложения (*reuse*)

- Съдържание

- оценка на степента на риска при различни приложения -> данни от тестовете

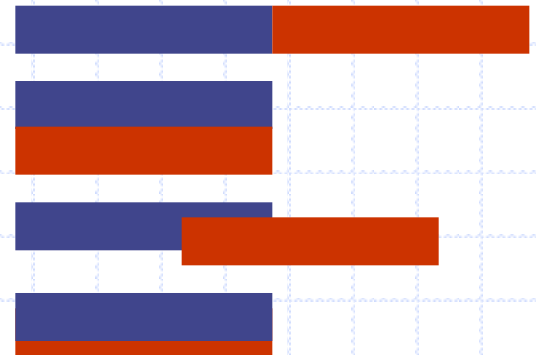
- дефиниране на функциите на БД -> документация на моделите и схемите

- определяне на отговорностите по поддържане -> деклариране на собствеността и правата за ползване



Използване и поддържане

- Изготвяне на документация
- Процедури за въвеждане в експлоатация
 - директно
 - паралелно
 - постепенно
 - пилотно



Използване и поддържане

- План за поддържане
 - Интерактивен и итеративен процес
 - видове поддръжка
 - при повикване
 - периодично
 - on-line
- План за обучение на потребителите

Оценка

- Цел: установяване на съответствие с изискванията
- Задачи:
 - дефиниране на система от критерии за оценка
 - установяване на недостатъците
 - анализ на ефективността на поддържане
- Резултат: води до първия етап от жизнения ЦИКЪЛ

Информационни системи



методологии за проектиране и
реализация



Методология за създаване

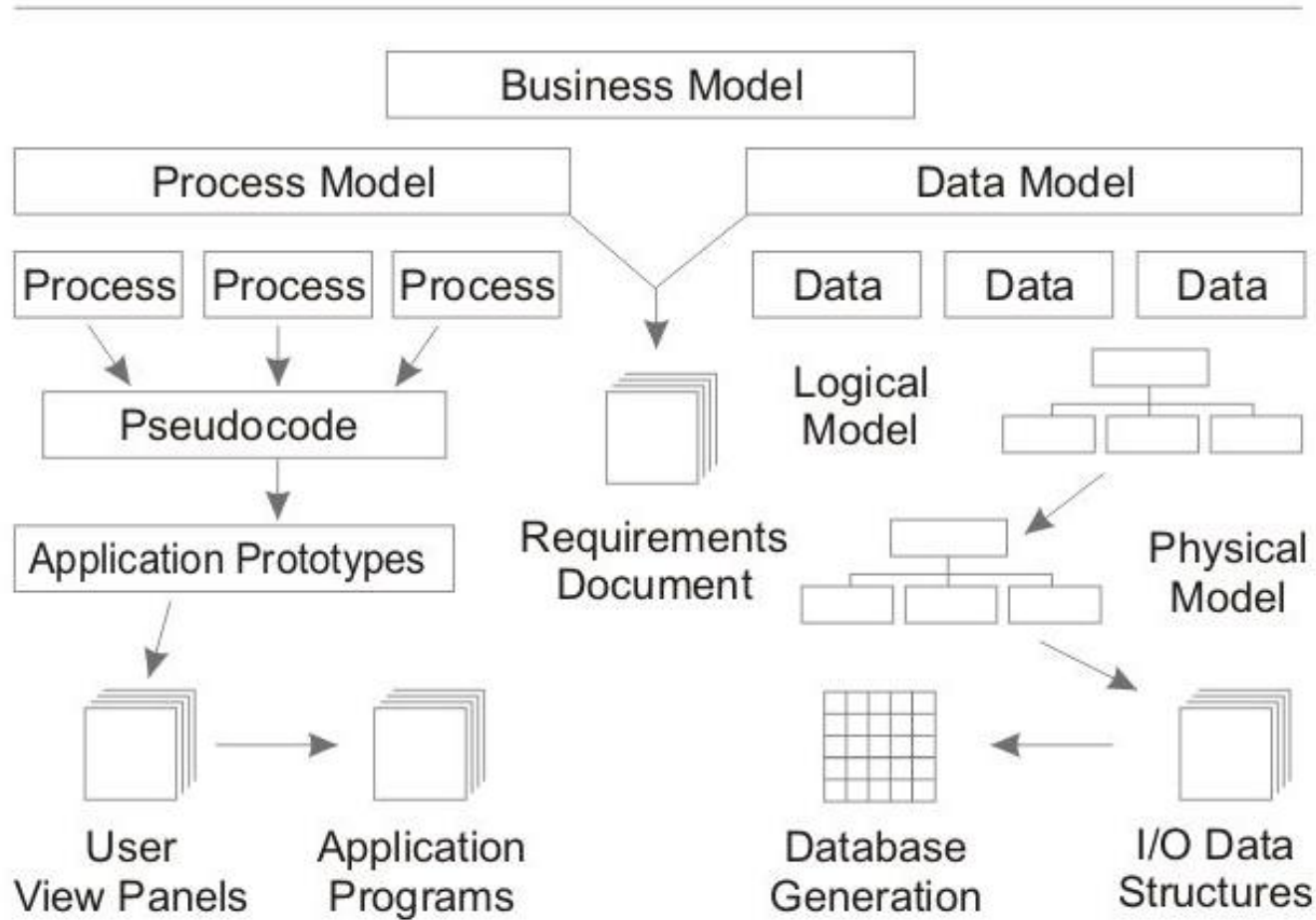
- Методология – формален подход за реализиране на жизнения цикъл
- Колекция
 - от методи,
 - организирани в последователност
- Конкретната реализация зависи от приоритета на
 - бизнес-процесите
 - данните

Подходи

- Основани на процесите
 - Основни обекти за процесите в бизнеса
 - Системата се представя като множество от процеси, към и от които тече информация
- Основани на данните
 - Основни обекти са данните, тяхната организация и съхранение
 - Системата се представя като структура, чието ядро са модели на данните
- Обектно-ориентирани методологии
 - Балансиране на фокуса между процесите и данните
 - Системата се представя колекция от обекти, които отразяват данните и процесите едновременно (UML)

Интегриран подход

Business Model Integration



Видове методологии

- Две категории
 - Планирани – предварително дефинирани изисквания и процедури за удовлетворяването им
 - Адаптивни – предварително дефинирани изисквания и адаптивни (настройваеми, не безпланови!) процедури за удовлетворяването им
- Реално методологиите се разполагат между двете крайности

<-Модулни-> <--Итеративни--> <--Водопад-->

<-----|-----|-----|----->

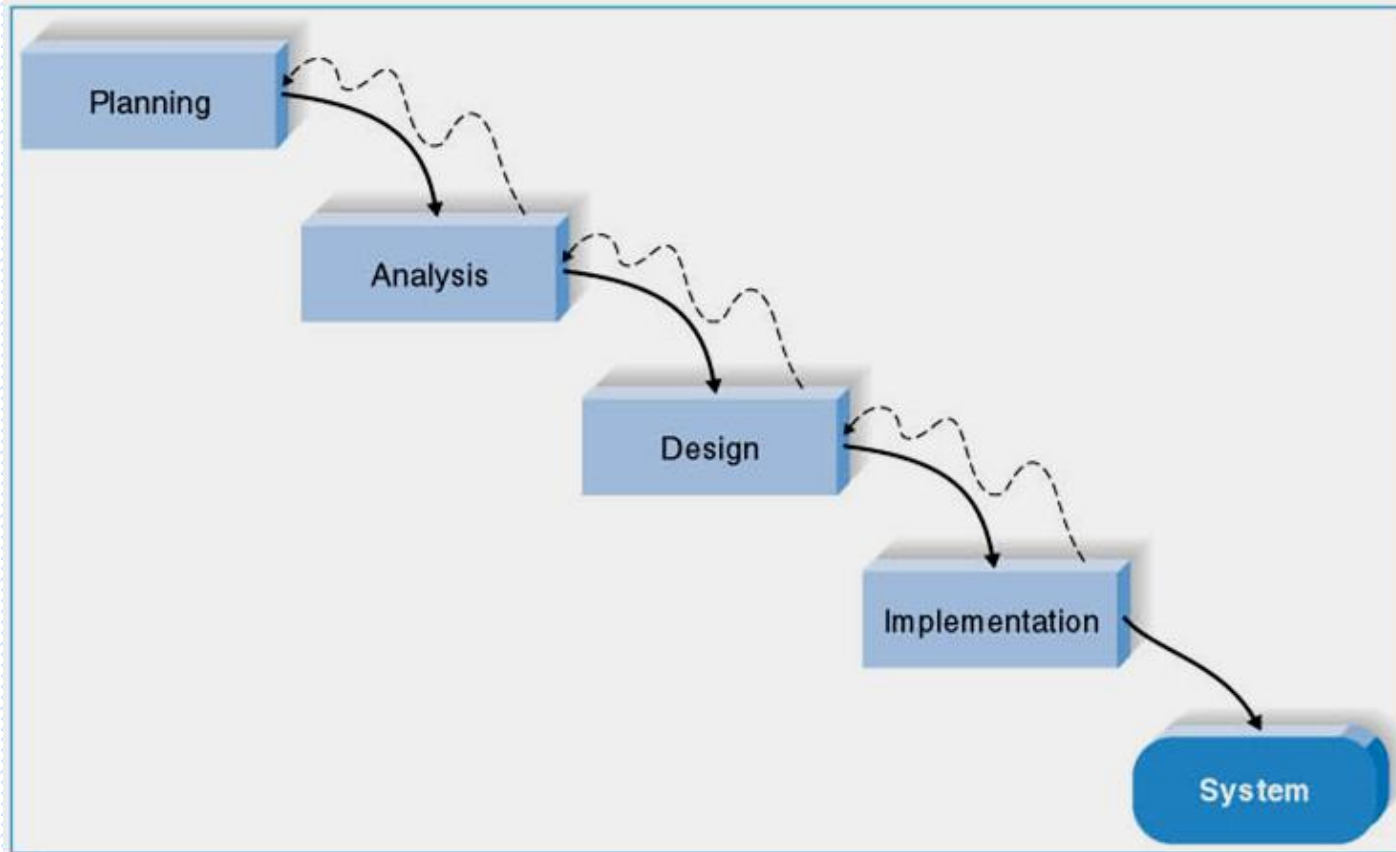
Адаптивни

Планирани

Видове методологии

- Структурно проектиране
 - стъпка по стъпка
 - формални модели
 - диаграми
 - предимства
 - предварително-дефинирани изисквания към системата
 - недостатъци
 - проектирането трябва да е напълно завършено преди програмирането
 - голям интервал от време между иницирането на проекта и прилагането му
- Реализации
 - Waterfall
 - Structured Systems Analysis and Design Method – SSADM, Soft Systems Methodology, Yourdon Structured Method и др.

Waterfall Development-based Methodology Водопад



Водопад

- Линеен и последователен метод
- Необратим
- Потребителите и създателите си взаимодействат на всяка стъпка
- Предимства
 - изискванията са дефинирани далече преди програмирането
 - фрагментиране и контрол
 - без итерации
 - след започване на програмирането се допускат малки промени
- Недостатъци
 - трудно усъвършенстване

SSADM

- Реализация на Waterfall
- Използвани техники
 - логическо моделиране
 - идентифициране, моделиране и документиране на съществени за бизнеса обекти и отношения
 - моделиране на потоците от данни, които се движат в информационната система
 - моделиране на поведението на обектите - идентифициране, моделиране и документиране на събитията, които влияят на обектите и начина на реакция

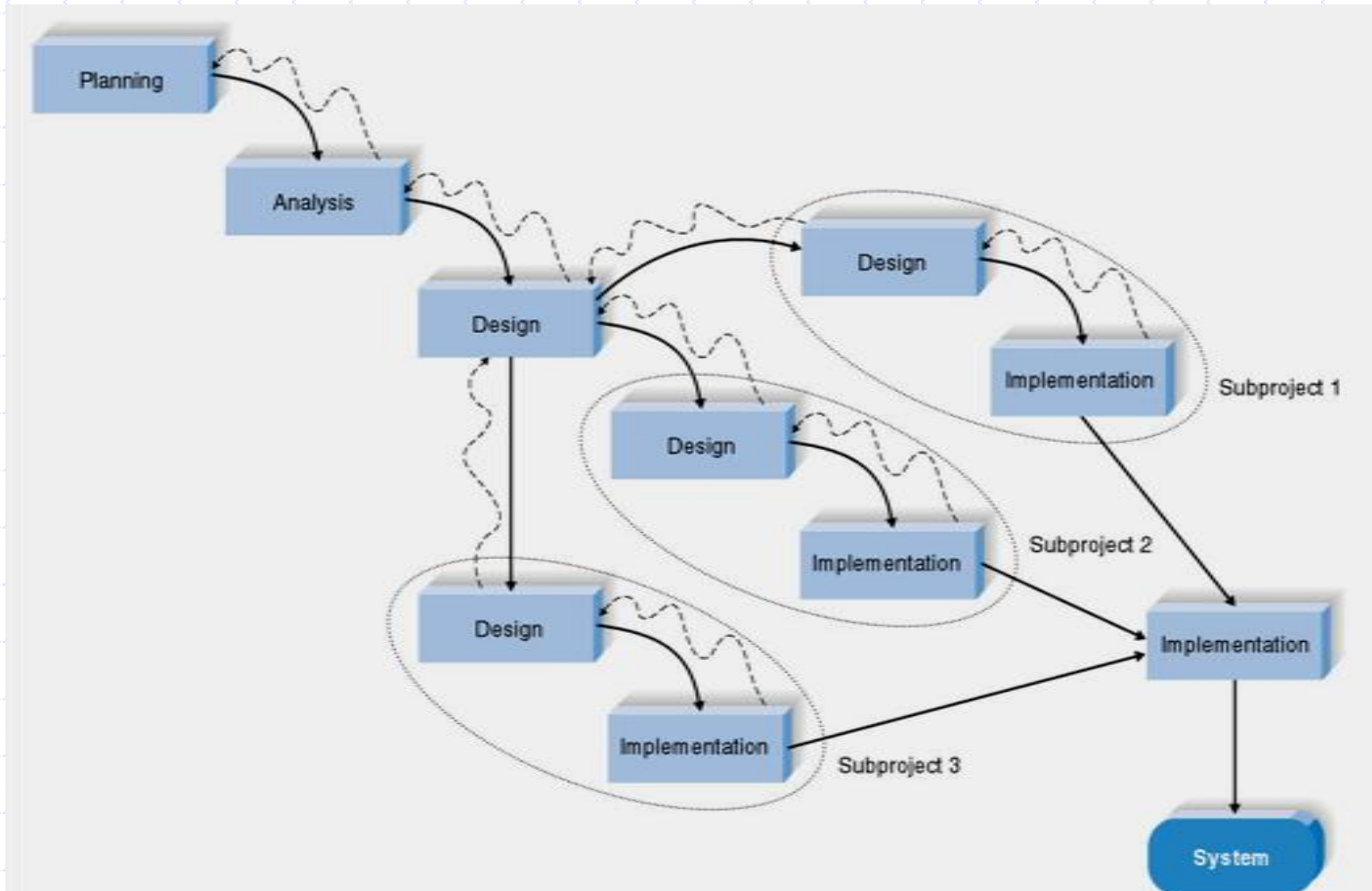
SSADM

- Етапи
 - Изследване на възможностите за реализация
 - Изследване на текущата практика
 - Допустими варианти за бизнеса
 - Дефиниране на изисквания
 - Допустими варианти за техническо решение
 - Логическо проектиране
 - Физическо проектиране

Паралелна методология

- За намаляване на интервала между етапите за анализ и приложение
- Прави се анализ и проектиране на цялата система, след това задачата се разделя на малки подзадачи (проекти)

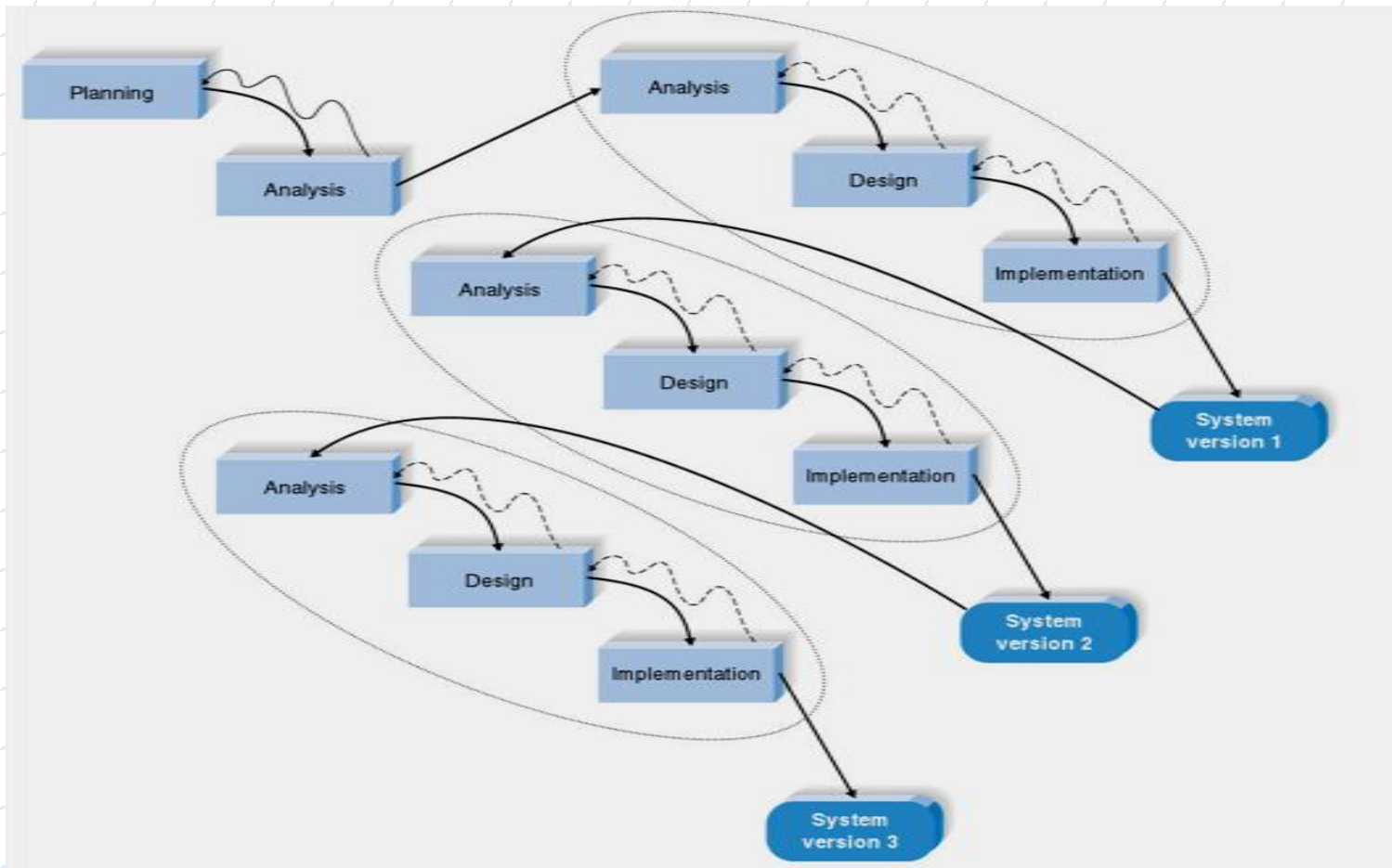
Паралелна методологија



Итеративна методология

- Създаване на фази
- Системата се представя чрез набор от последователни версии
- Във всяка версия се използват версии на изискванията: най-важните – най-напред
- Прави се анализ и дефиниране на изисквания, но проектът и продуктът от първа версия удовлетворяват само част от изискванията
- След завършването на една версия започва следващата

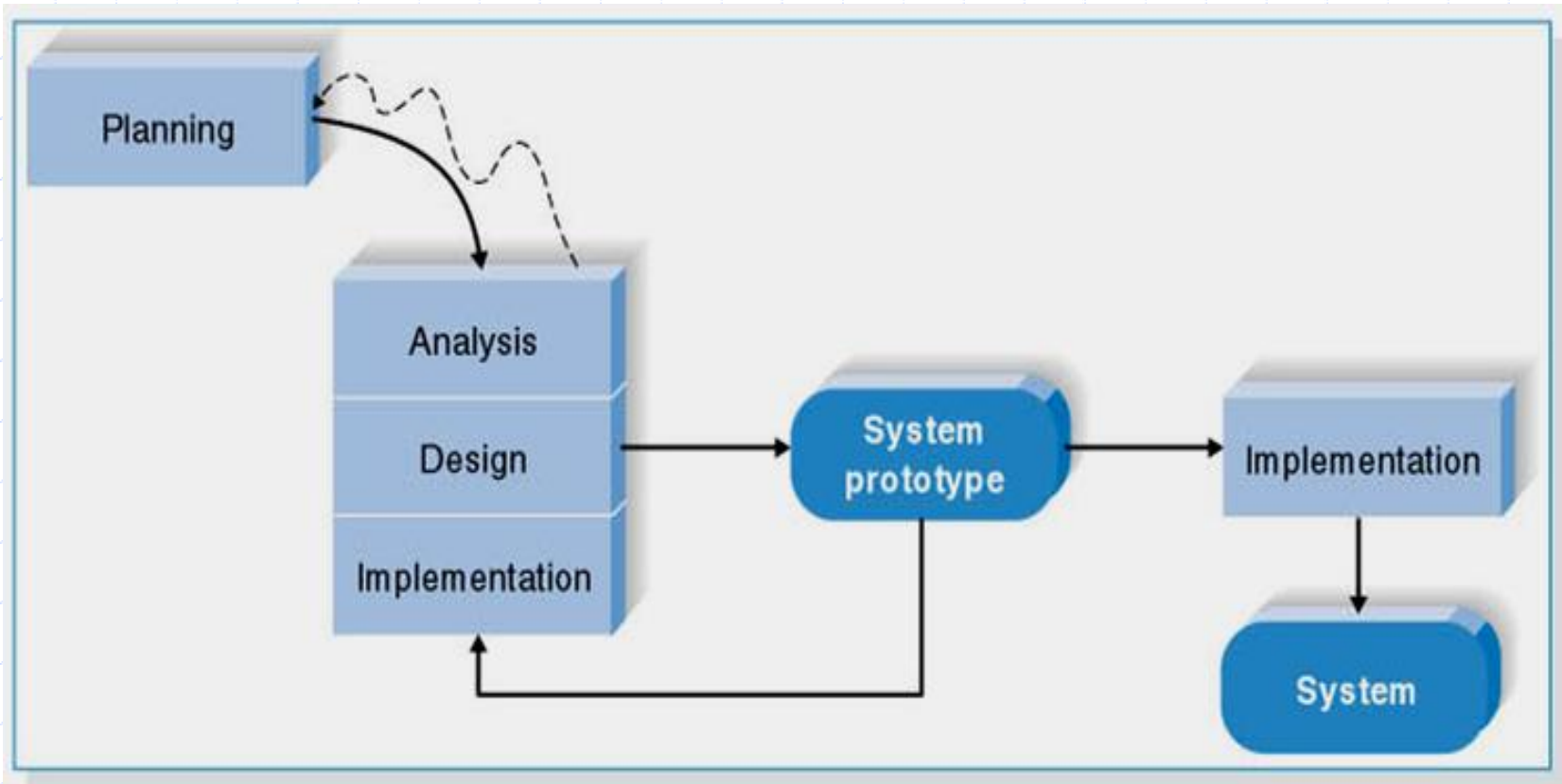
Итеративна методология



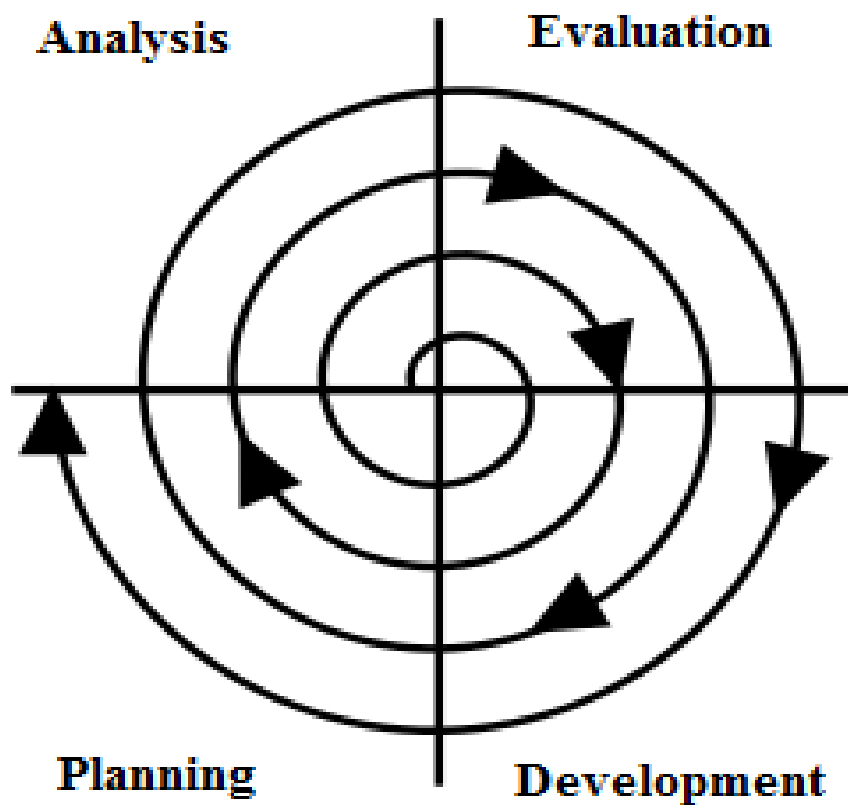
Методология, основана на прототипи

- Създаване на прототип: построяване на модел на системата, която ще бъде построена
- Посредством анализ, проектиране и програмиране се създават прототипи (скелет)
 - прототипите се дават на потребителите за използване
 - при използването се осъществяват промени на модела
 - скелетът се допълва, напр. добавят се процедури за верификация и др.
- Трите фази се повтарят до окончателното завършване на системата
- Предимство: потребителите могат да пробват системата преди завършването ѝ
- Недостатък: Липсата на начален цялостен проект води до грешки

Методология, основана на прототипи



Спираловиден подход



Agile - “пъргав” подход

- Семейство от съкратени методологии
- Ускоряване на процеса посредством елиминиране на части от моделирането и документирането
- Опростено **итеративно** и **инкрементално** създаване
- Представители
 - Rapid Application Development RAD
 - Extreme programming XP
 - Dynamic Systems Development Method DSDM

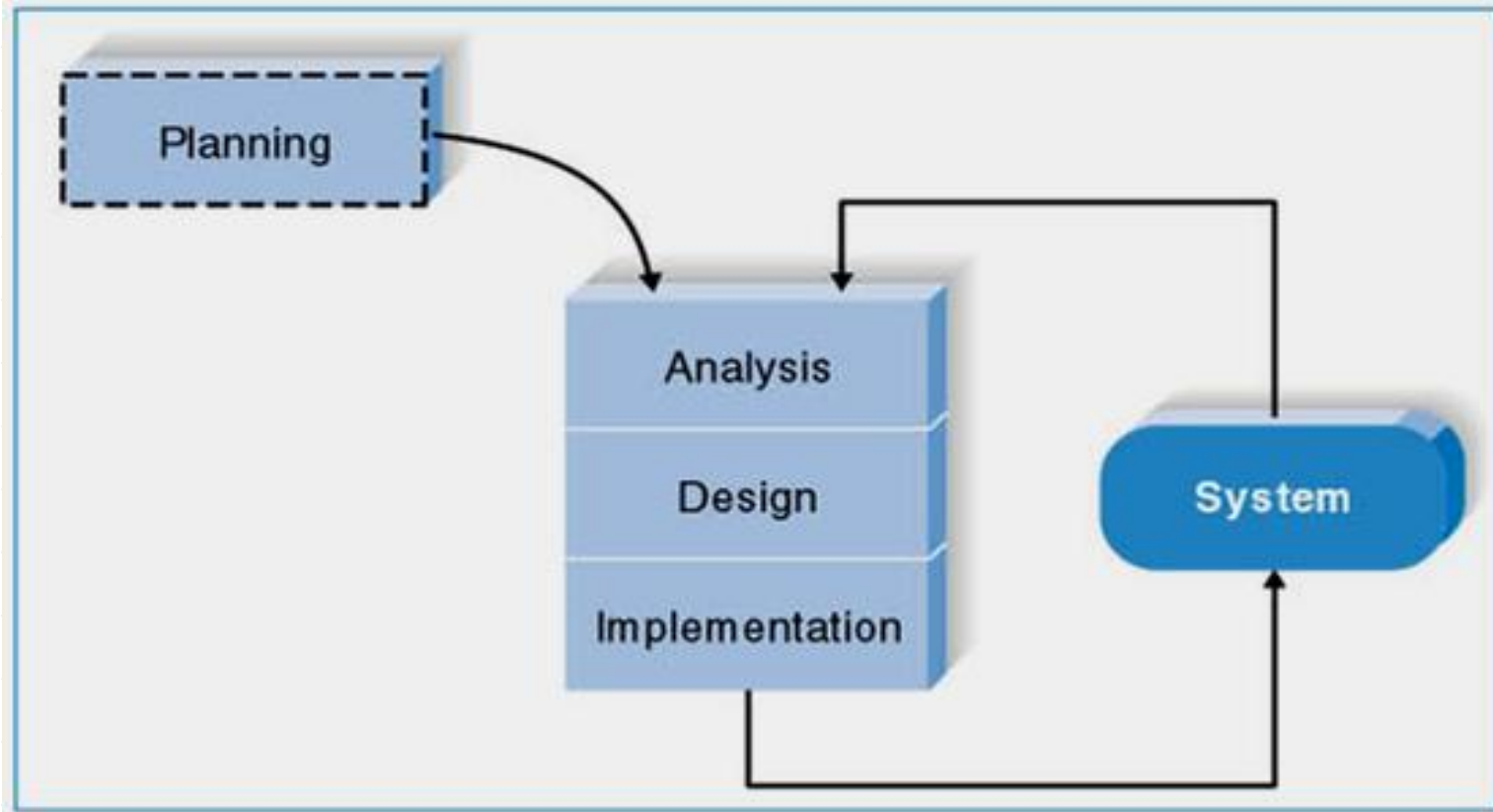
Rapid Application Development (RAD)

- Основна цел: дефиниране на изисквания
- Отделни части от жизнения цикъл се завършват по-бързо и се предоставят на потребителите
- Минимално планиране, съчетано с програмирането
 - модел на данните и бизнес процесите
 - бързо изготвяне на прототип
- Използват се специални техники и CASE за ускоряване на процесите при анализ, проектиране и внедряване
- Допълнителен проблем е удовлетворяването на нарастващите и изменящи се изисквания на потребителите

Extreme Programming (XP)

- 5 основни начина за постигане на ефективност
 - комуникация
 - простота
 - обратна връзка
 - респект
 - смелост
- 3 основни принципа
 - непрекъснато тестване
 - опростено кодиране
 - близки контакти с потребителите за бързо завършване

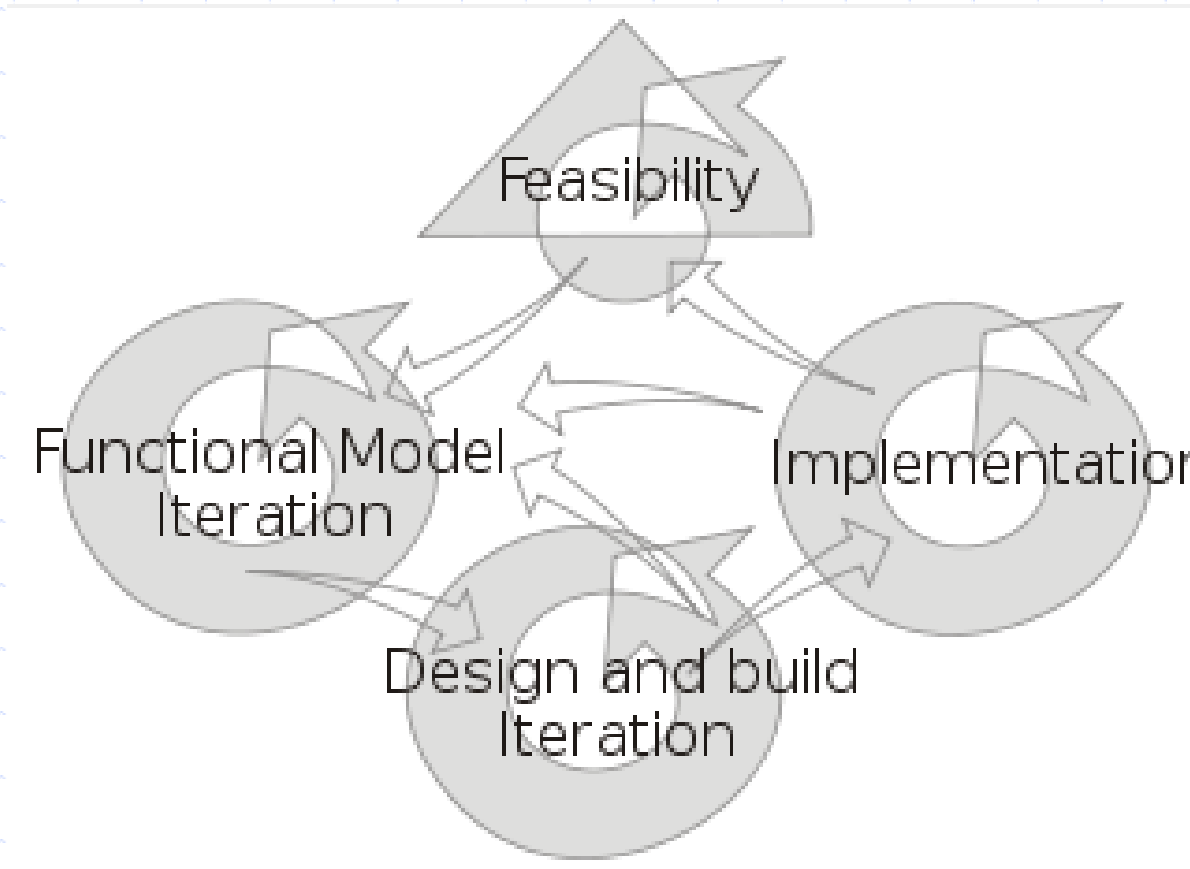
XP Methodology



Dynamic Systems Development Method DSDM

- След 2007 – основен метод за създаване на проекти
- Итеративен и инкрементален подход
- Непрекъснатата координация между потребители и създатели
- Цели се спазване на
 - сроковете
 - бюджета
- за сметка на спецификациите
- Последна реализация: DSDM Atern

DSDM



Етапи на DSDM

- Изследване
 - на възможностите за реализация
 - на бизнеса
- Итеративно построяване на функционален модел
 - определяне на функционален прототип
 - определяне на график
 - създаване на прототипа
 - оценка на прототипа
- Итеративно проектиране и реализация на модела за безопасно използване от потребителите
- Приложение
 - одобрение на реализацията от потребителите
 - обучение на потребителите
 - тестово приложение
 - контрол и документиране

Избор на подходяща методология

- Труден
- Зависи от стандартите на бизнеса

Ability to Develop Systems	Structured Methodologies			RAD Methodologies		Agile Methodologies
	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	XP
with Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent
with Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Poor
that are Complex	Good	Good	Good	Poor	Excellent	Poor
that are Reliable	Good	Good	Good	Poor	Excellent	Good
with a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent
with Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Good

Избор на подходяща методология

- Потребителски изисквания
 - RAD - при неясно дефинирани изисквания
- Познаване на технологиите
 - за адекватност между очакванията и възможностите
- Сложност на системата
 - сложните системи изискват методология, включваща детайлен анализ
- Надеждност на системата
 - системи с висока надеждност изискват методология, включваща детайлен анализ
- Продължителност на разработката
 - RAD – подходящи за бързи разработки
 - Waterfall – неподходящи, поради невъзможност да се планират промени в графика
- Възможност за оценка на риска в началото на проектиране
 - RAD