

ДИЗАЙН НА СОФТУЕРНИ СИСТЕМИ

Софтуерна архитектура. Обектно-ориентиран дизайн.

СУ „Св. Климент Охридски“
Факултет по Математика и Информатика
Увод в Софтуерното Инженерство

Въведение

- Дотук изяснихме следните понятия
 - Софтуерно инженерство
 - Процес за разработка на софтуер

Процес за разработка на софтуер

Анализ на изискванията
(Requirements)



```
graph TD; A[Анализ на изискванията (Requirements)] --> B[Дизайн (design)]; B --> C[Реализация (Implementation)]; C --> D[Валидация и верификация (Testing, Verification, Validation)]; D --> E[Поддръжка (Maintenance)];
```

Дизайн (design)

Реализация
(Implementation)

Валидация и верификация
(Testing, Verification, Validation)

Поддръжка
(Maintenance)

Понятие за дизайн

- Колко бързо може да се справите със следните задачи:
 - Метод на Java, който събира две числа
 - Програмна реализация на калкулатор?
 - С графичен интерфейс
 - В конзолен режим
 - Програма за личен календар
 - Сървър за електронна поща
 - Социална мрежа (Facebook)

Понятие за дизайн

- (Създаване на) план или практика за създаване на обект или система в рамките на дадена среда, като се използва набор от базови компоненти и се удовлетворяват дадени изисквания
- Всички големи (сложни) системи се нуждаят от дизайн
 - Софтуерните системи не са изключение

Примери за сложни/големи системи



Всички системи се нуждаят от дизайн



А какво означава „система“?

Колко пъти сте виждали това?



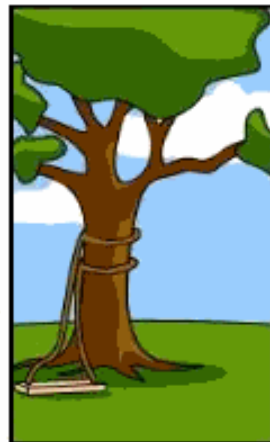
How the customer explained it



How the Project Leader understood it



How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



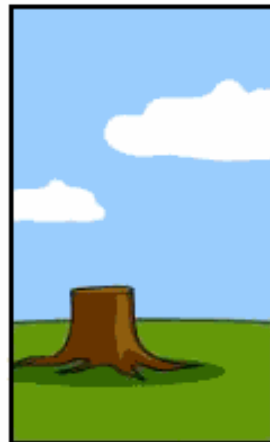
How the project was documented



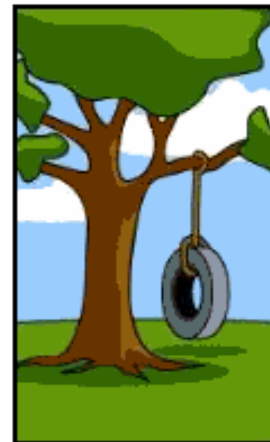
What operations installed



How the customer was billed



How it was supported



What the customer really needed

Софтуерното инженерство (по света и у нас)

Какво иска (клиента)
потребителя



Как го обяснява



Софтуерното инженерство (по света и у нас)

Какво разбира ръководителя на
проекта



Какъв дизайн се прави

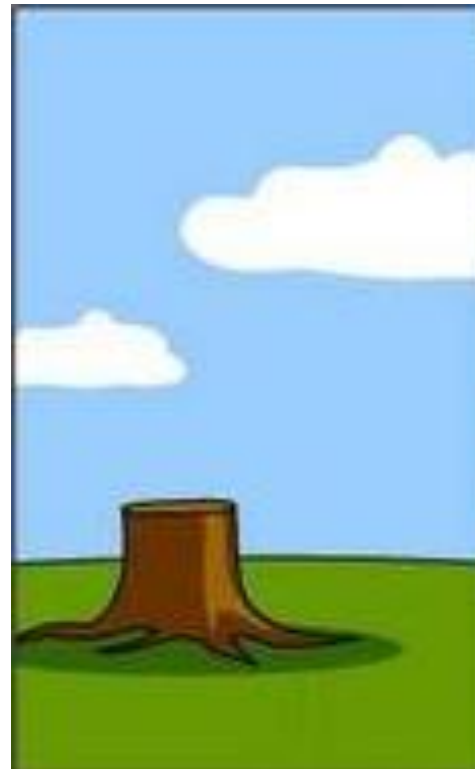


Софтуерното инженерство (по света и у нас)

Какво пише програмиста

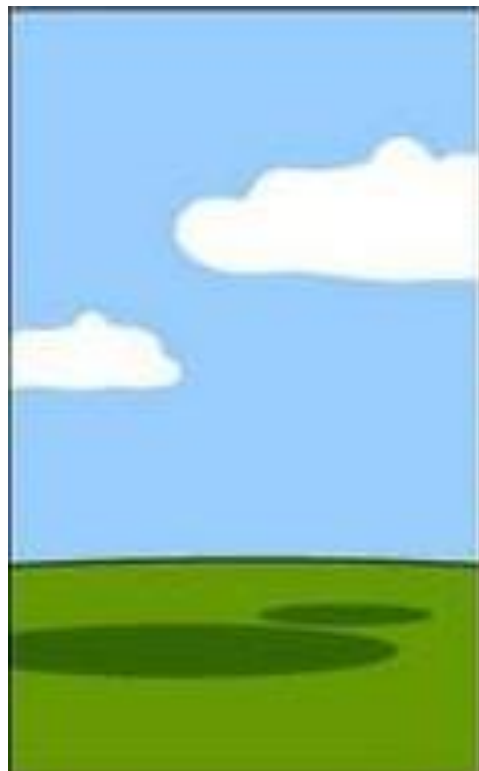


Каква поддръжка е осигурена



Софтуерното инженерство (по света и у нас)

Какво е документирано за
продукта



Каква цена е поискана от
клиента



Какви изводи може да се направят

- Комуникацията между различните участници в разработката е трудна
- Необходим е общоприет начин за описание и комуникация между заетите в разработката на различни аспекти на софтуерните системи през всички фази на разработка

- В софтуерното инженерство фазата на дизайн се отъждествява с проектиране на т.нар. архитектура на софтуерната система.

Софтуерна архитектура

- Дефиниция
 - Съвкупност от различни структури на системата, които се състоят от софтуерни елементи, външно видимите характеристики на тези елементи, и връзките, които съществуват между тях

[Bass, Clements & Kazman, 2003]

- Визуализацията на дадена структура се нарича изглед (view)

Софтуерна архитектура

- Всъщност, като всяко понятие в софтуерното инженерство, съществуват най-различни дефиниции и мнения за това какво представлява архитектурата
- Независимо от това, тя представлява основна и много важна концепция в софтуерното инженерство

What is Software Architecture?

- **Definition:**

- A software system's architecture is the set of *principal design decisions* about the system

- Software architecture is the blueprint for a software system's construction and evolution

- Design decisions encompass every facet of the system under development

- Structure

- Behavior

- Interaction

- Non-functional properties

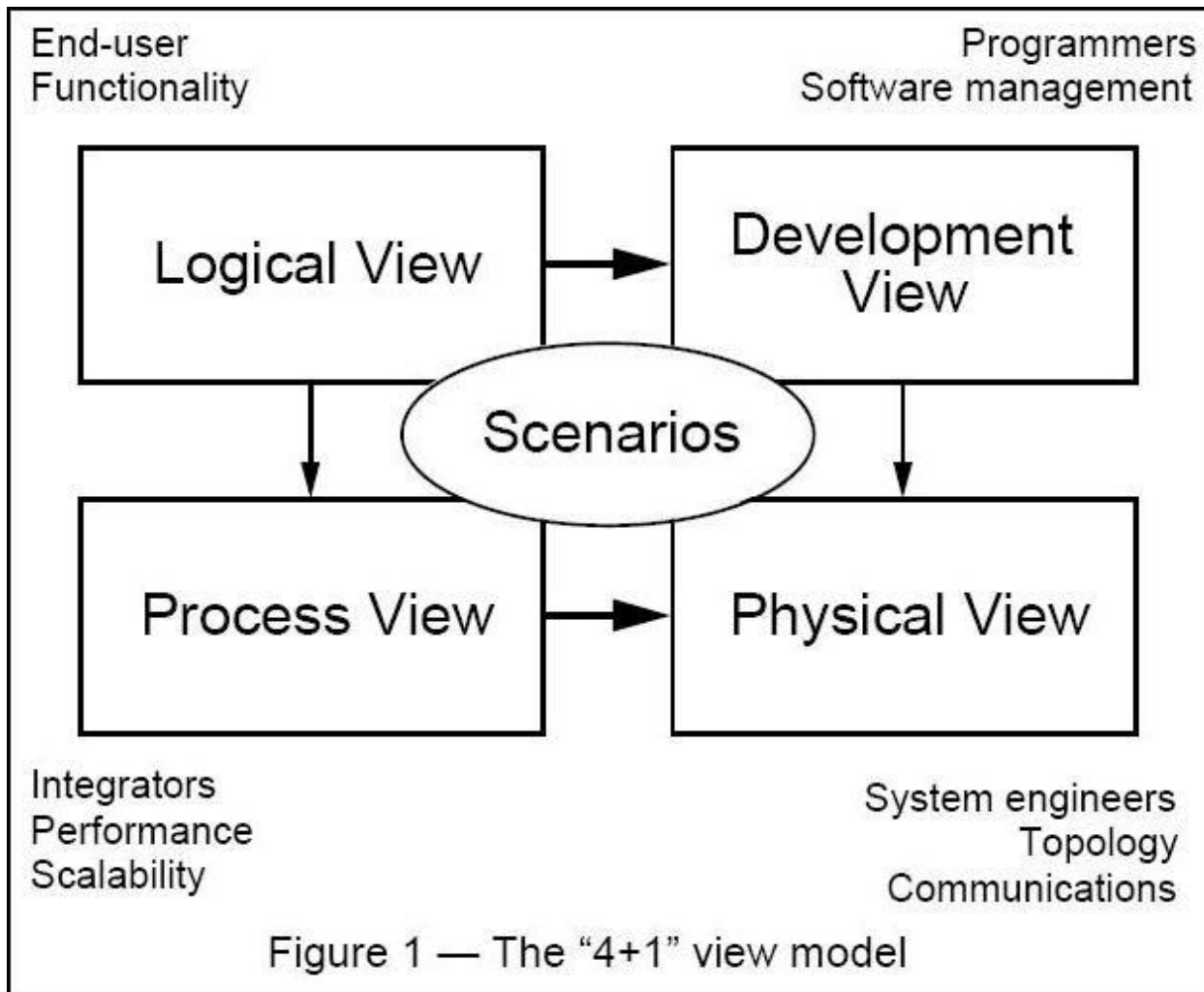
Software architecture

- The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design.
- The output of this design process is a description of the software architecture.

Софтуерна архитектура

- Различни дефиниции
 - Основните решения по дизайна на софтуерната система, които включват
 - Вътрешна конструкцияСтруктура
 - Поведение
 - Взаимодействия (вътрешни и с други системи)
 - Качествени характеристики
 - Подробно описание на конструкцията и начините за развитие на софтуерната система
 - И т.н.
- Около 50 дефиниции на понятието може да се намерят на:
 - <http://www.sei.cmu.edu/architecture/start/glossary/community.cfm>

4 + 1 модел на софтуерната архитектура



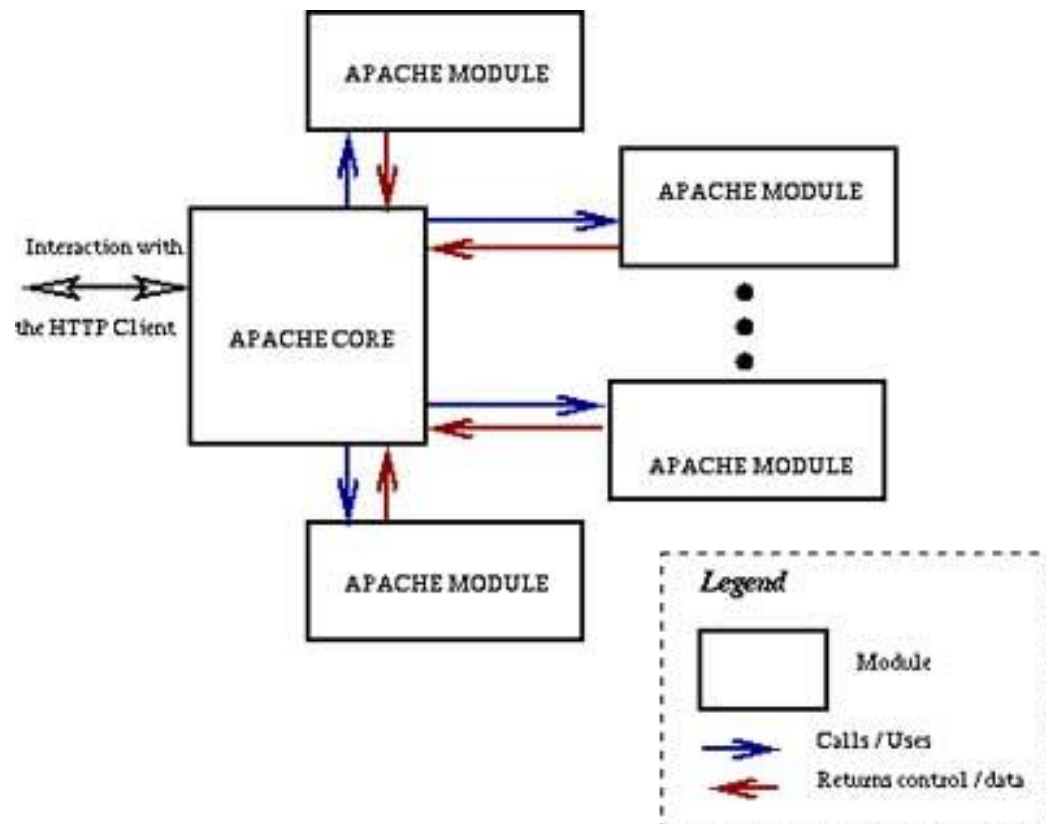
Kruchten, Philippe B. "The 4+1 view model of architecture." *Software, IEEE* 12.6 (1995): 42-50.

4 + 1 модел на софтуерната архитектура

- 1) Логически изглед – показва основните абстракции в системата, като обекти, класове и компоненти
 - 2) Изглед на процесите – показва системата като съвкупност от взаимодействащи си процеси по време на изпълнение
 - 3) Изглед на кода – Показва как отделните елементи на системата се разполагат във файлове код
 - 4) Физически изглед – показва как софтуерните компоненти са разпределени между хардуерните възли в системата
- +1) Съответните сценарии на употреба

Какво означава структура/изглед

А кои са елементите на една структура?



Модулна структура

- Елементи на архитектурата
 - Изчислителен модул (component)
 - Съединителен модул - CM (connector)
 - Обвивен модул (wrapper)
- Архитектурна конфигурация
- Архитектурен стил
 - Често използвани и доказали се в практиката, конкретни архитектурни конфигурации

Външно видими характеристики на даден компонент

- Това са нещата, които другите компоненти знаят за него. Например:
 - Услуги, които той предоставя
 - Качествени характеристики
 - Информация за изключения и обработка на грешки
 - Употреба на споделени ресурси
 - И т.н.
- Дефинират се от т.нар. интерфейси
 - Входен интерфейс – описва какво е необходимо на компонента, за да извърши работата си
 - Изходен интерфейс – описва какво компонентът ще извърши, ако са изпълнени всички условия на входния му интерфейс

```
Int compute (int num1, int num2,  
string mode)
```

Дизайн на архитектурата

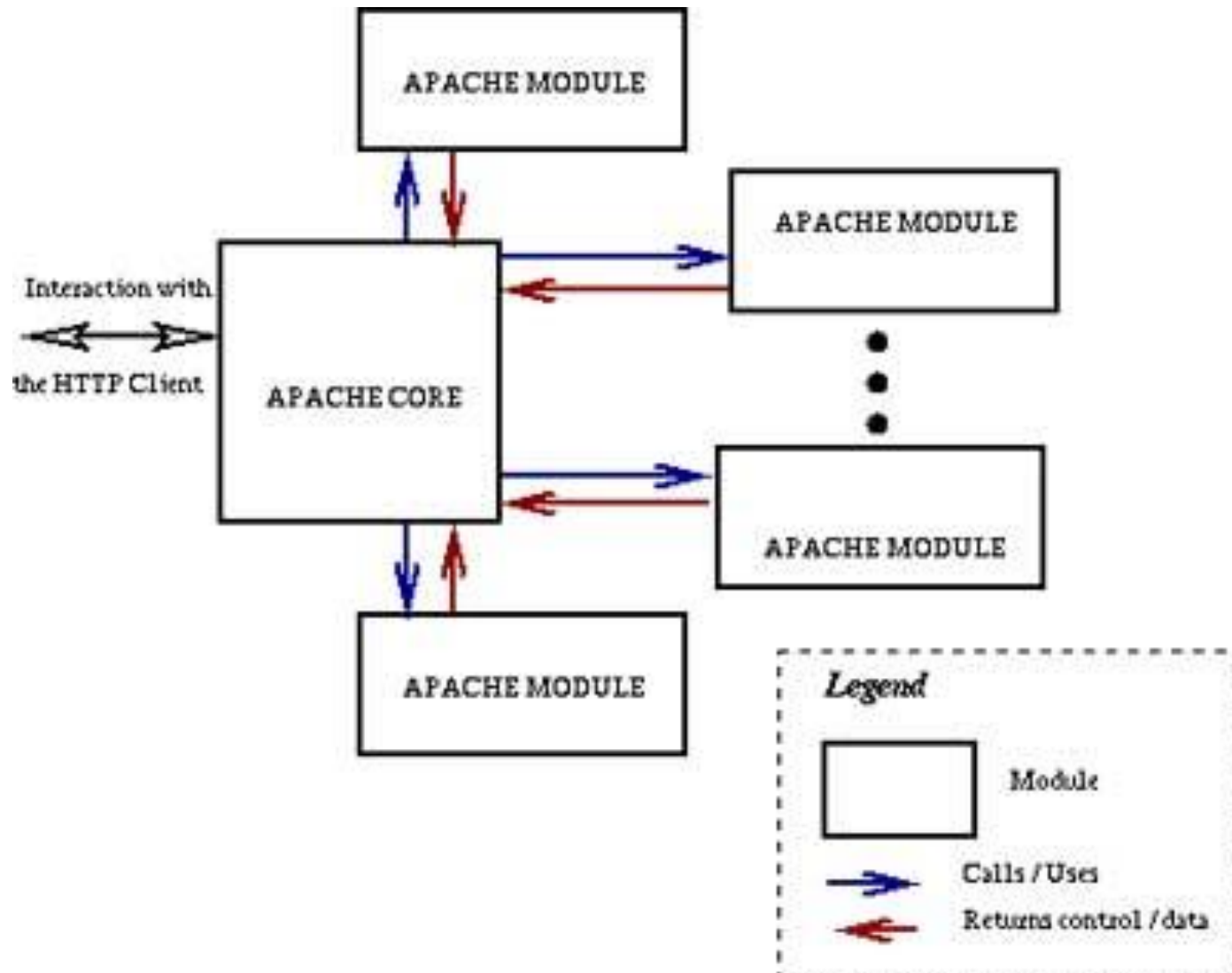
- Процесът на дизайн на архитектурата в по-голяма степен представлява съвкупност от решения отколкото последователност от някакви конкретни действия

Архитектурен дизайн

- Процесът на проектиране, при който се идентифицират подсистемите на една система и начина по който те комуникират и биват управлявани в рамките на системата
- Резултатът от процеса на архитектурен дизайн е документация (***описание на архитектурата***)

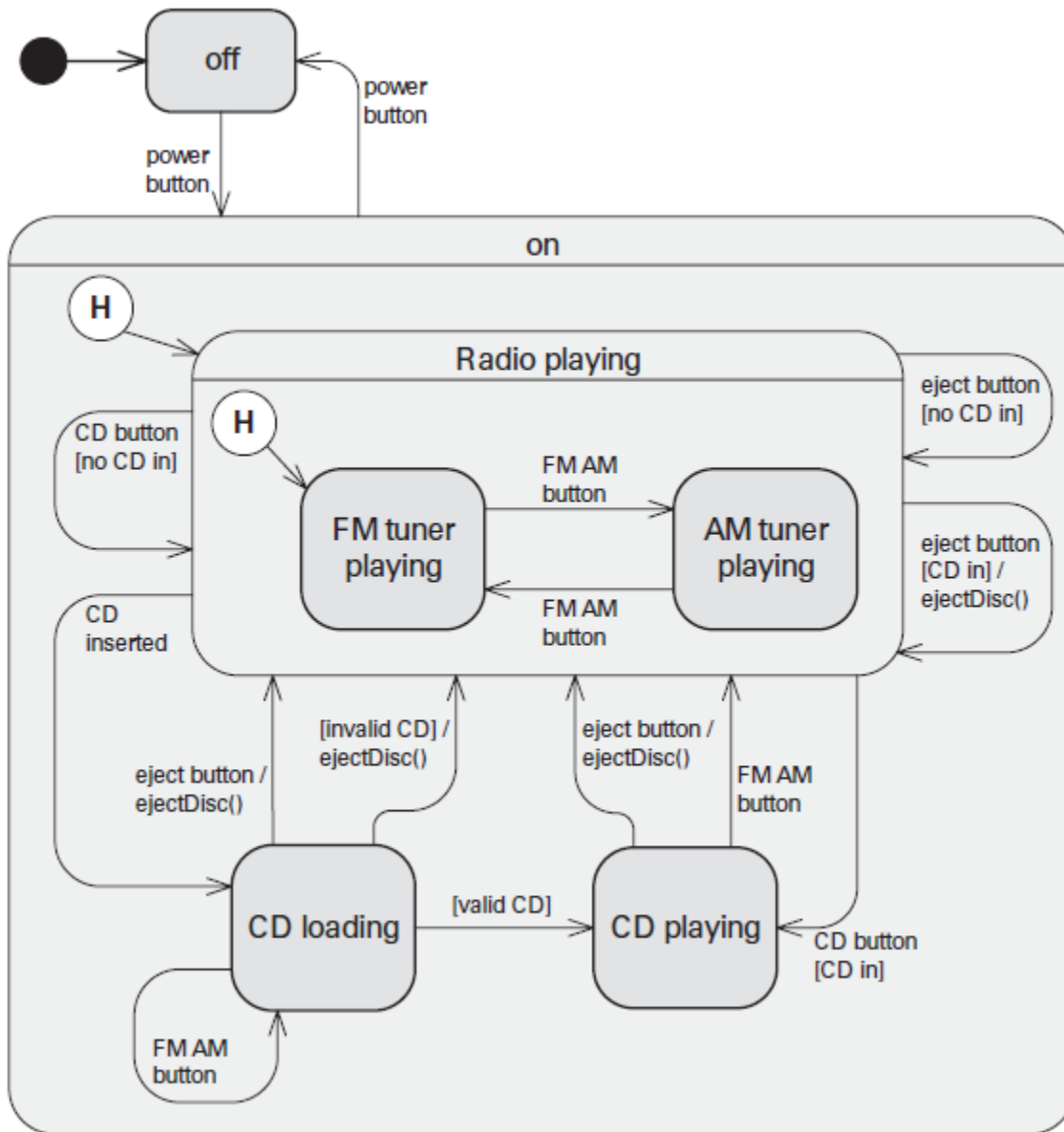
Описание на архитектурата

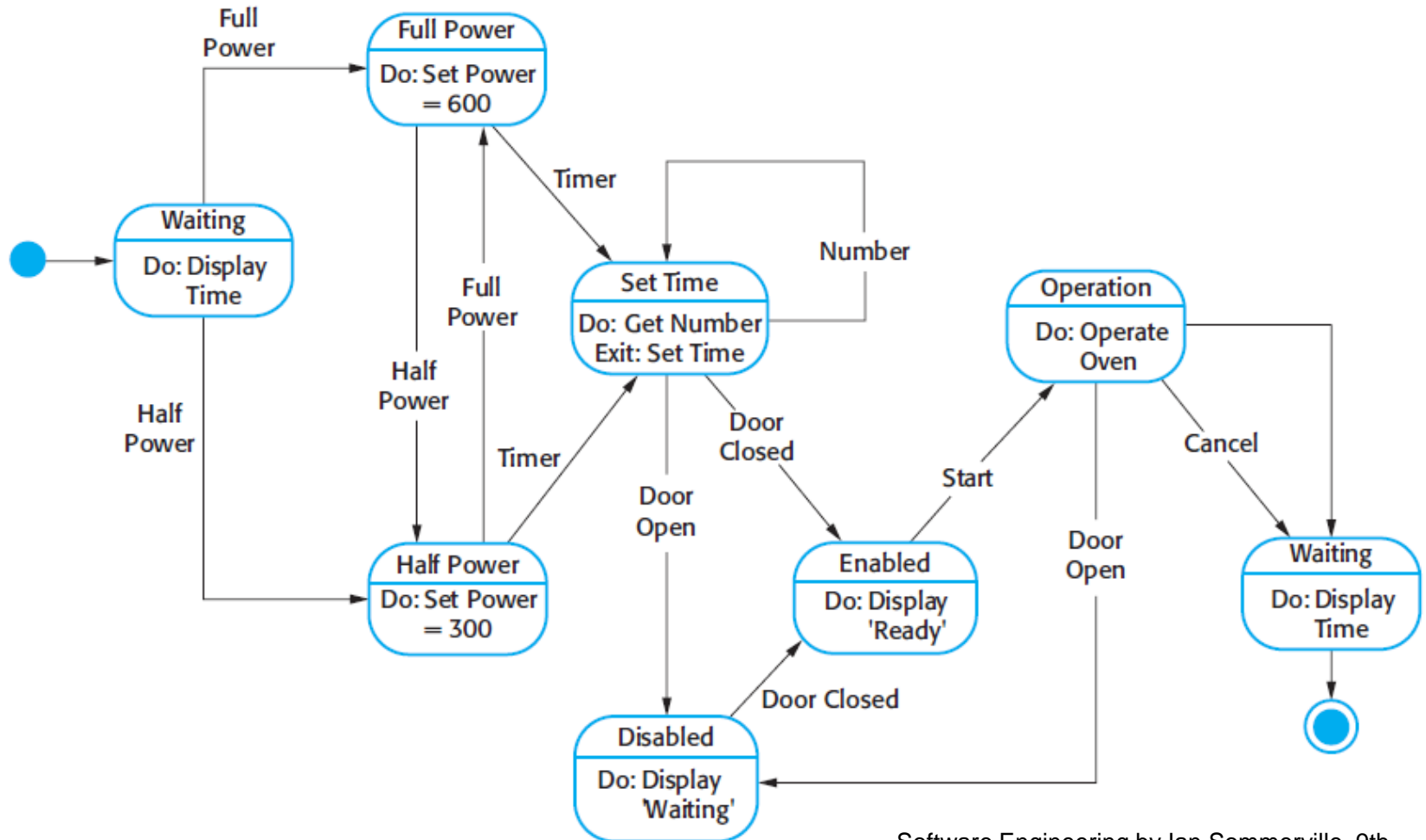
- Документация, модел, описание?
 - Кое от трите?
 - Всички заедно?
- Защо е нужен модел на софтуерната система?

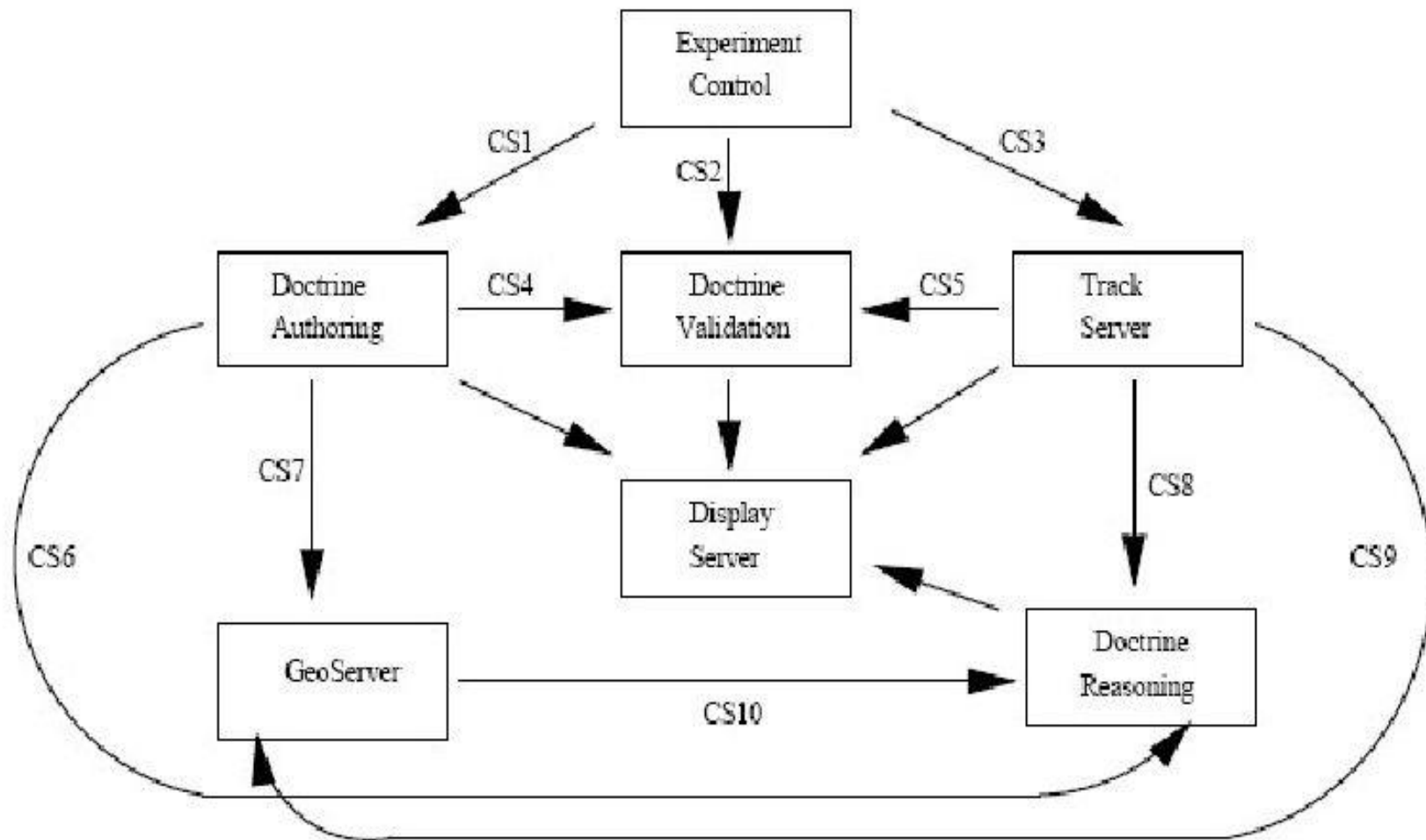


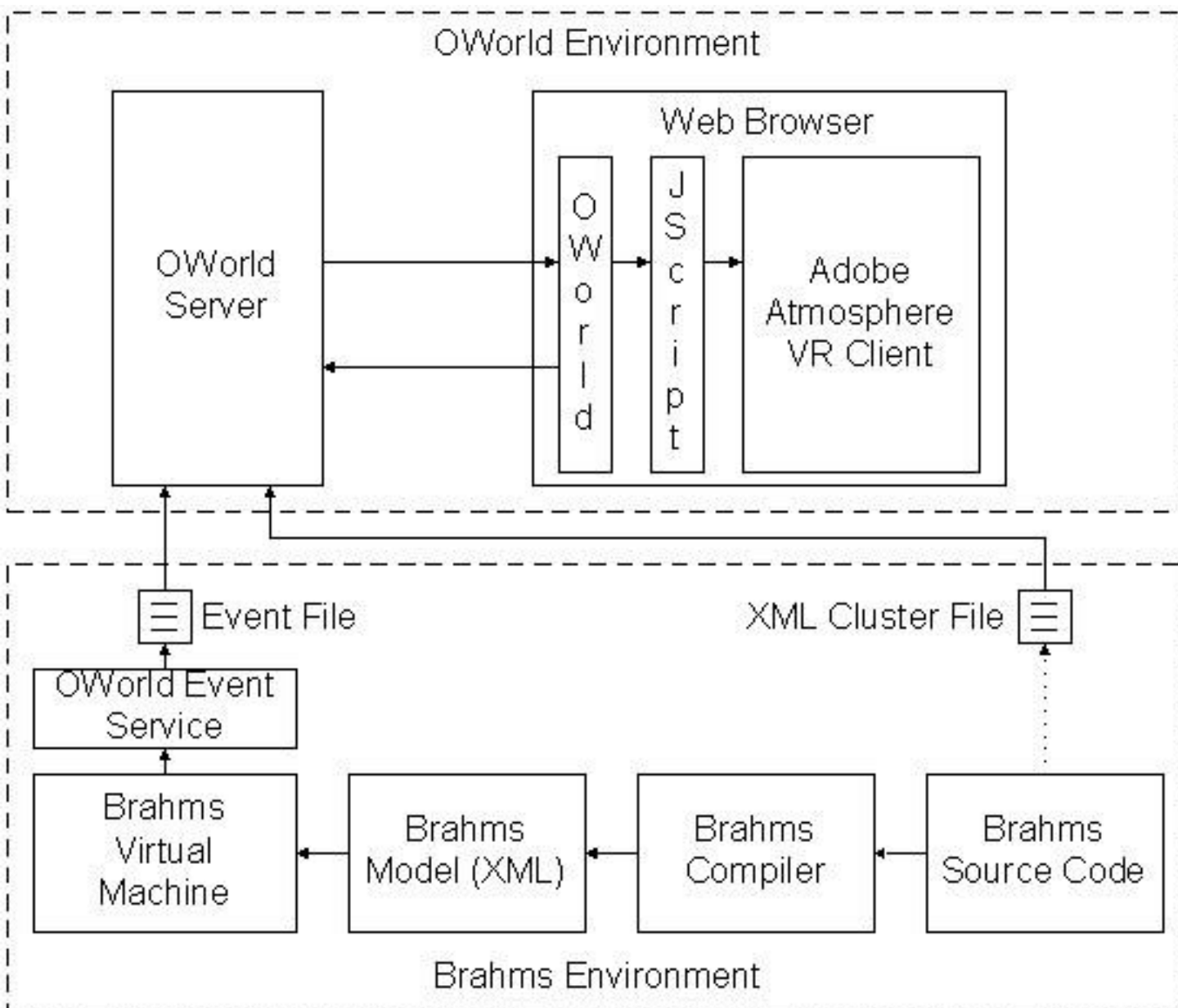
Какво наричаме описание на архитектурата

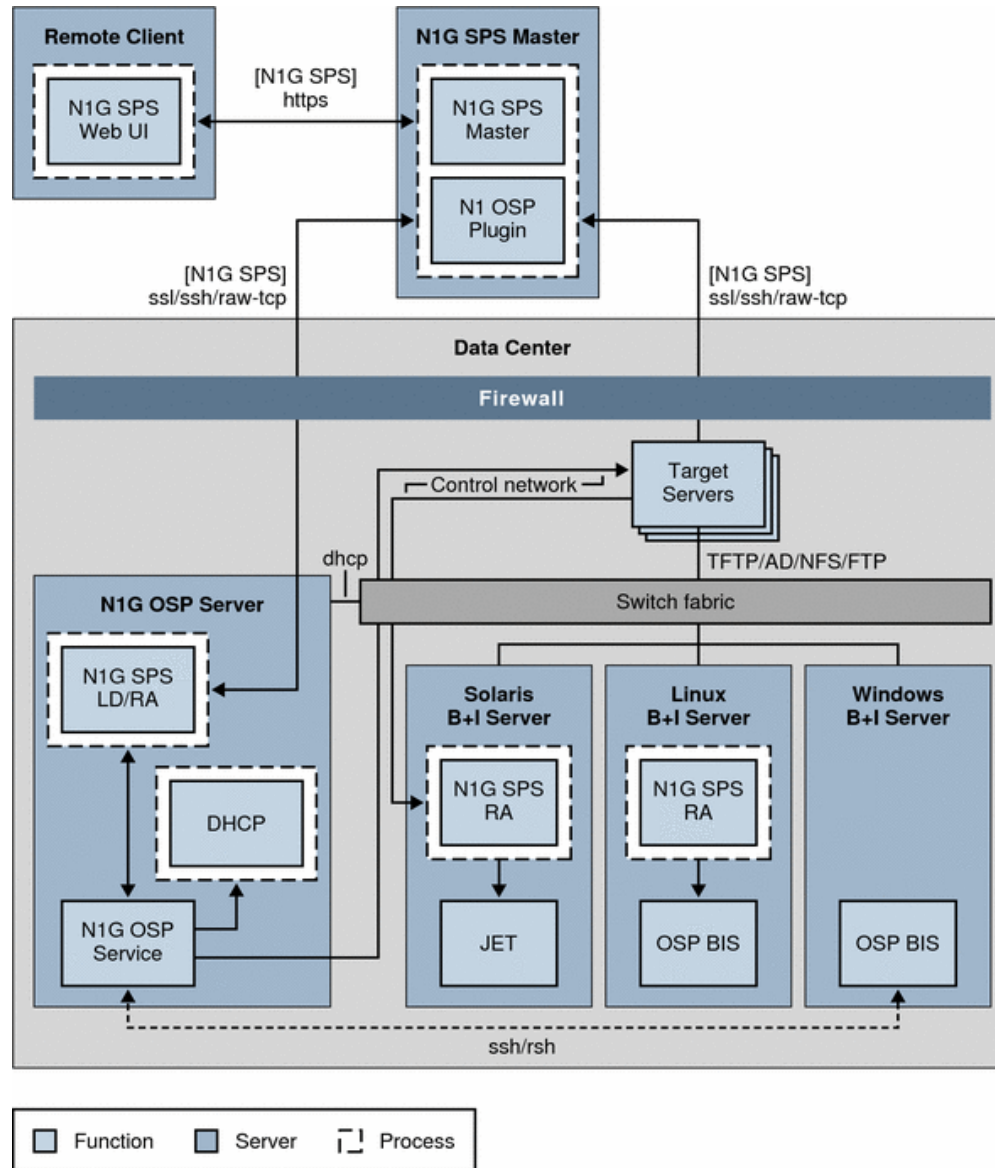
- Описание от тип “кутии и линии”
 - Твърде абстрактно – не показват нито последователността на взаимодействието между компонентите, нито дават повече информация за зависимостта между подсистемите (и компонентите) и т.н.
 - Въпреки този недостатък, този тип описания са полезни за комуникация между участващите в проекта лица, тъй като те имат различно ниво на технологични познания

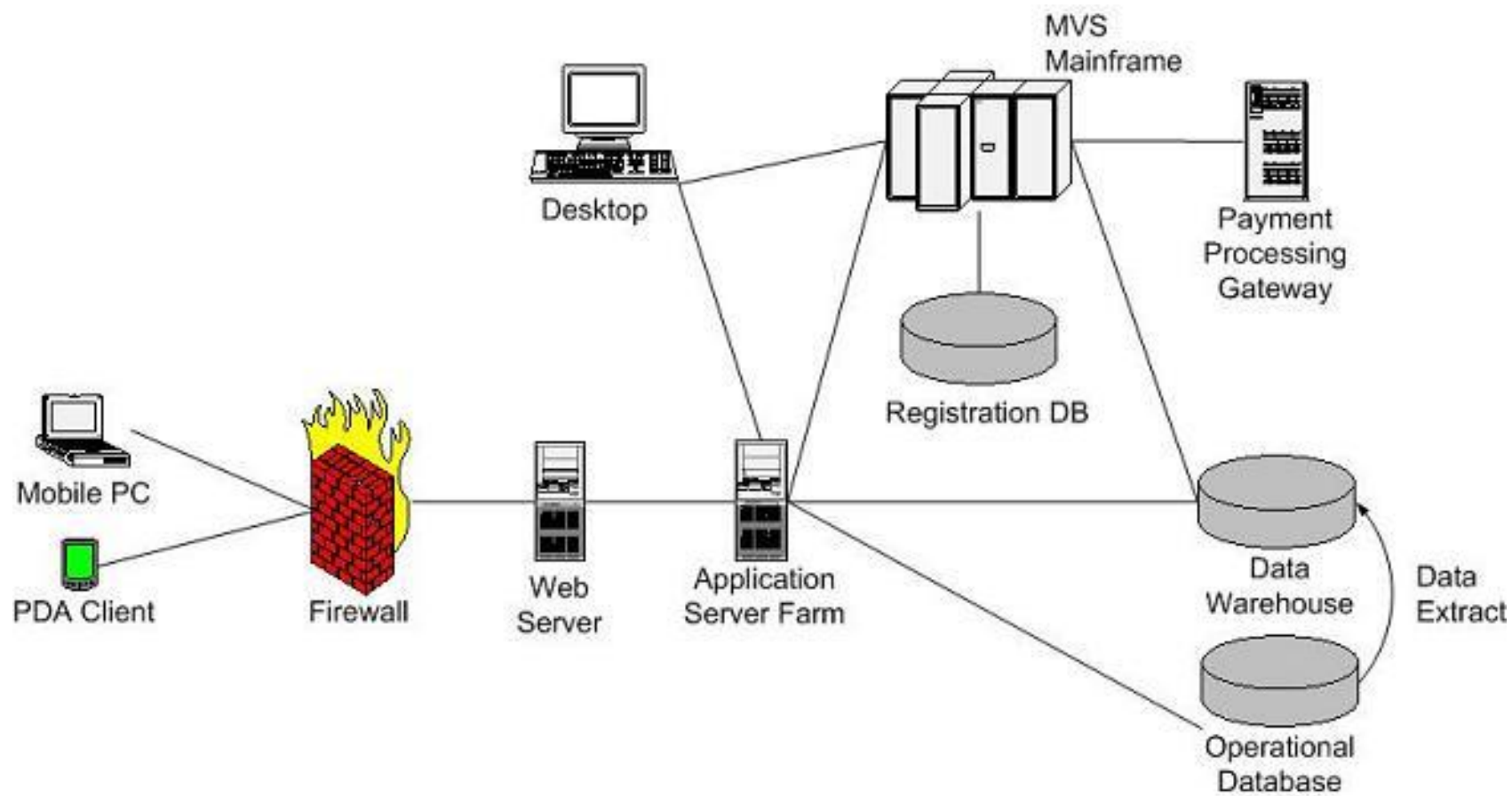












Възможна употреба на моделите на архитектурата

- Като начин за улесняване на комуникацията и дискусията относно дизайна на системата
 - Липсата на много технически детайли в даден абстрактен изглед (view) към системата го прави полезен, както за комуникация между **заинтересованите лица**, така и за улесняване на планирането на проекта за разработка на системата. Заинтересованите лица може да дискутират системата, без да се затормозяват с излишни детайли .
 - Специализираните изгледи, дават възможност за дискусии в рамките на специфични аспекти на архитектурата
 - Например: комуникация между компоненти и процеси, синхронизация между процеси, разположение на компонентите в разпределени системи и др.
- Документация с цел улесняване на бъдеща поддръжка на системата

Защо е нужно да се описва архитектурата

- Комуникация между заетите в проекта по разработка
- Анализ на системата
- Многократна употреба
- Видът на модела зависи от неговото предназначение

Най-често срещани проблеми при разработката на софтуер

- Расте броя на
 - Потребителите
 - Приложните области (наука, забавления, масови комуникации, медицина и т.н.)
 - Програмистите
 - Необходимост от взаимодействие между различни софтуерни приложения
- Всичко това води до
 - Голяма сложност на разработваните програми и системи
 - Трудна поддръжка
 - ...

Софтуерната архитектура е абстракция

- А какво означава абстракция?
 - Опростяване на общата картина, чрез скриване на определени детайли

Абстракция



Абстракция

- Нека проектираме клас „човек“, за два вида софтуерни системи
 - Система за управление на ресурсите в университет
 - Система за обучение по биология на деца до 5 г.

- Архитектурата е абстракция на софтуерната система, която „премълчава“ различни детайли за нея
 - Програмен език за реализация
 - Код и структура на кода
- От значение са детайли, които имат отношение към:
 - Взаимодействие между компонентите: *как те се използват* или *как те използват* други компоненти
 - Как компонентите се свързват помежду си
 - Как (в каква последователност) си взаимодействат
 - По какъв начин си взаимодействат с хардуера

Всички системи ли имат архитектура?

- **Абсолютно** всички имат архитектура
- Може би никой не е правил дизайн на системата преди тя да бъде разработена, но това не означава, че няма архитектура
- Също така може да липсва документация на архитектурата, което пак не означава, че няма архитектура
- Много често поради факта, че липсва документирано описание на архитектурата, се налага да се прави т.нар. reverse engineering

Reverse engineering

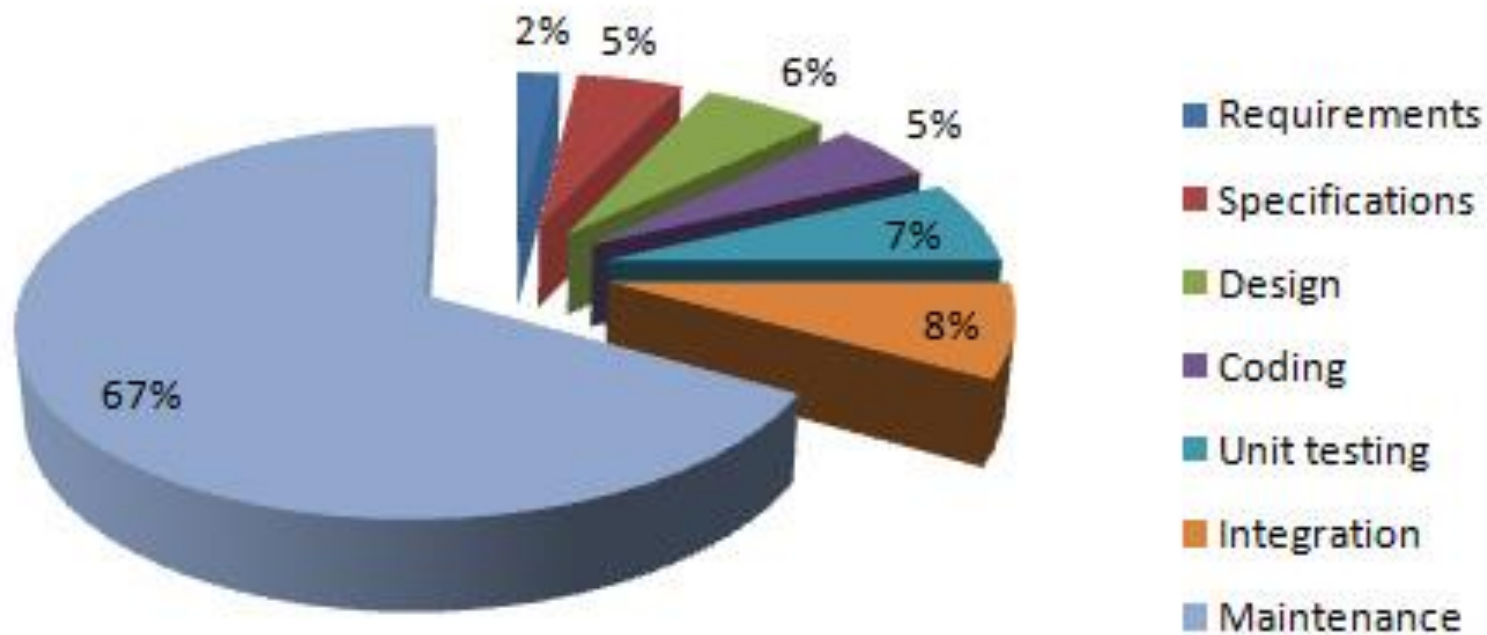
- Процес при който даден софтуер се анализира, за да се разбере структурата му
- Има голямо количество софтуер, който може да прави това автоматично
- Въпреки това много често е нужна човешка намеса
 - Времеемко
 - Скъпо

Наследени (legacy) системи

- Наследените системи са разработени отдавна, но се използват и днес, като при това може да са особено важни за правилното функциониране на даден бизнес процес.
 - Много често се оказва че са реализирани на остарял програмен език или технология или пък използват други системи и платформи, които не се поддържат.
 - Също така за тях съществува невалидна документация или пък дори липсва такава.
- Въпреки това, съществуват редица случаи в които е много скъпо или рисковано такива системи да се разработват отново.

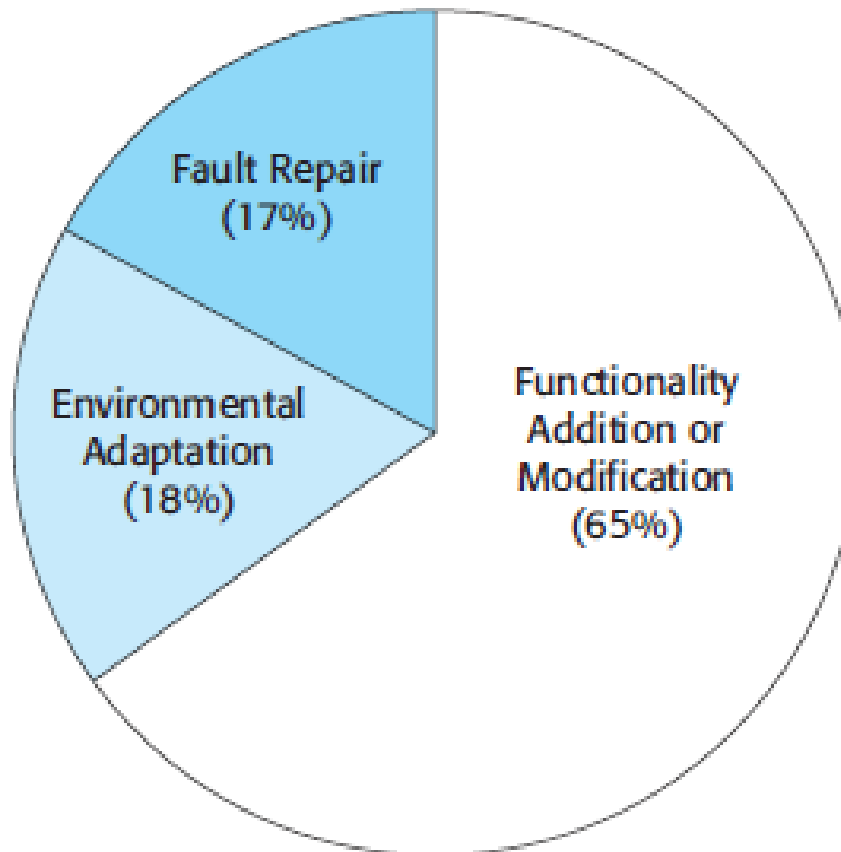
Разпределение на разходите за софтуер

Software Life-Cycle Costs



Source : Digital Aggregates

Разходи за поддръжка на системите



Обобщение

- Всички системи имат архитектура
- Ако подходим по-сериозно към нея
 - Улесняваме интеграцията
 - Подобряваме условията за многократна употреба
 - Улесняваме оперативната съвместимост между системите