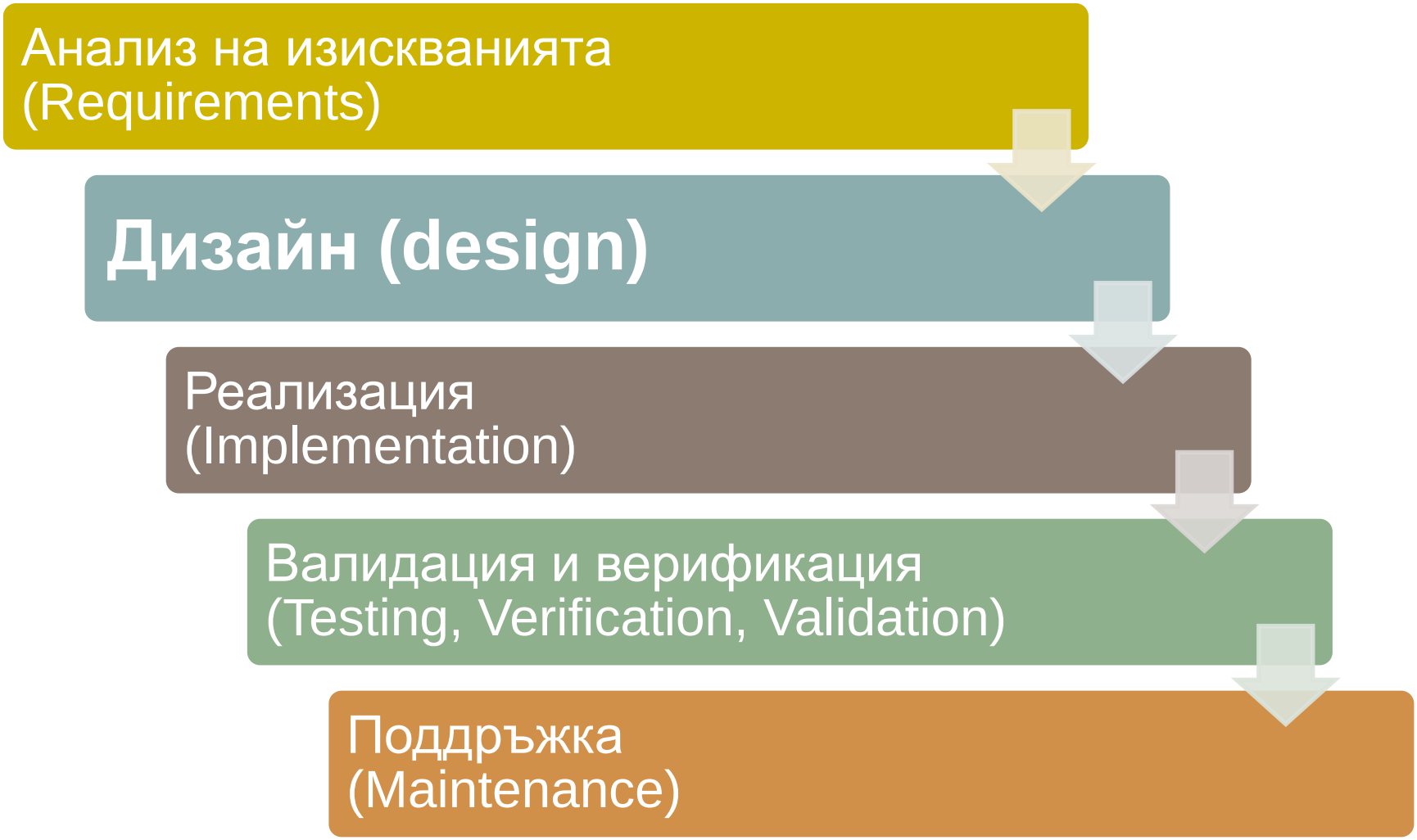
ДИЗАЙН НА СОФТУЕРНИ СИСТЕМИ

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
  - Описателни модели на софтуерен процес
  - Гъвкави методологии за разработване на софтуер
  - Анализ на изискванията към софтуера и процес на събиране

Процес за разработка на софтуер



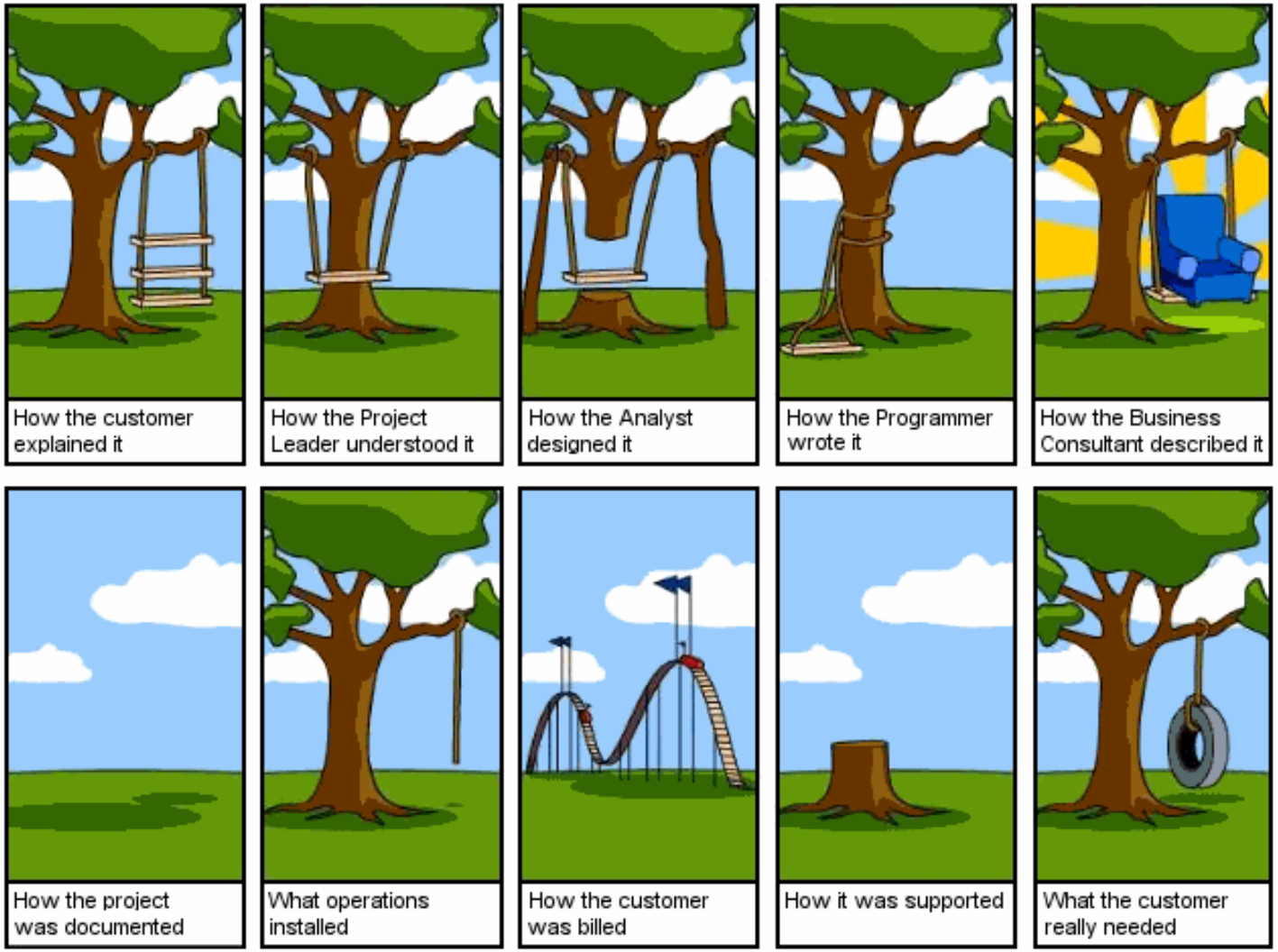
Понятие за дизайн

- (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

Колко пъти сте виждали това?



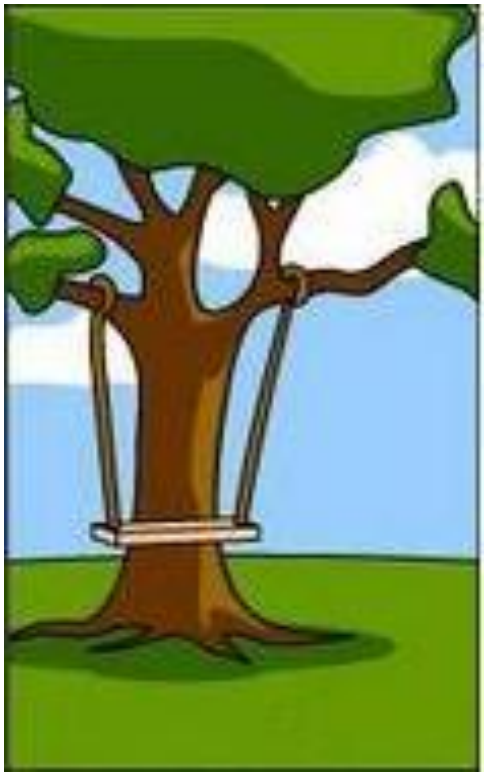
Софтуерното инженерство (по света и у нас)

Какво иска (клиента) потребителя

Как го обяснява

Софтуерното инженерство (по света и у нас)

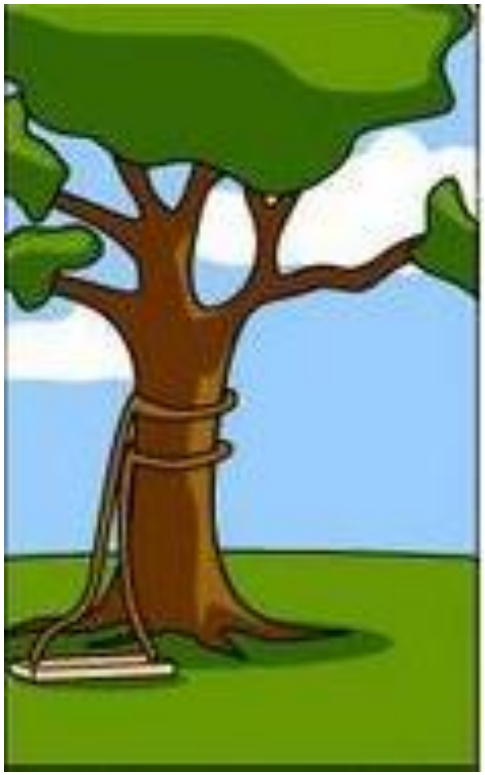
Какво разбира ръководителя на проекта



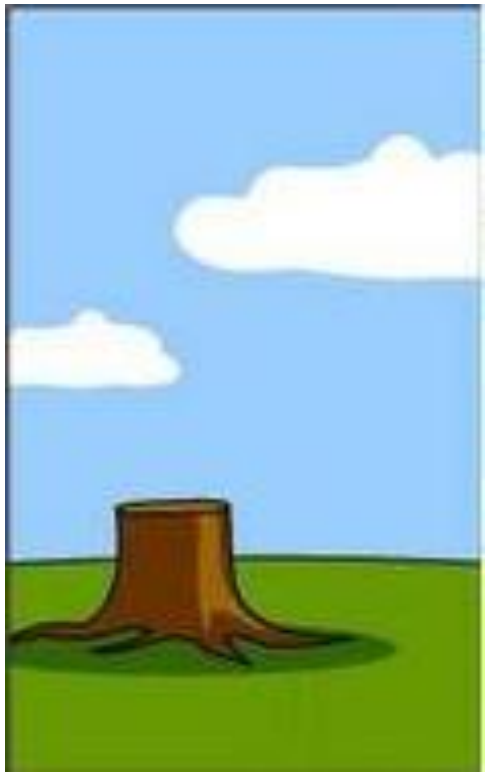
Какъв дизайн се прави

Софтуерното инженерство (по света и у нас)

Какво пише програмиста

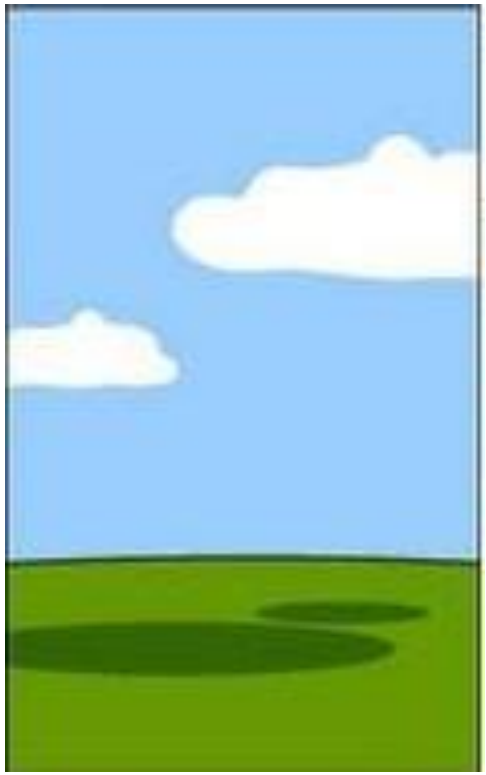


Каква поддръжка е осигурена



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

Какво означава софтуерна архитектура

- Архитектурата на софтуерната система е набор от основни дизайнерски решения за системата
- Софтуерната архитектура е планът за изграждането и еволюцията на софтуерната система
- Дизайнерските решения обхващат всеки аспект на системата в процеса на разработка
  - Структура
  - Поведение
  - Взаимодействие
  - Нефункционални свойства

Софтуерна архитектура

- Различни дефиниции
  - Основните решения по дизайна на софтуерната система, които включват
    - ВътрешнаконструкцияСтруктура
    - 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

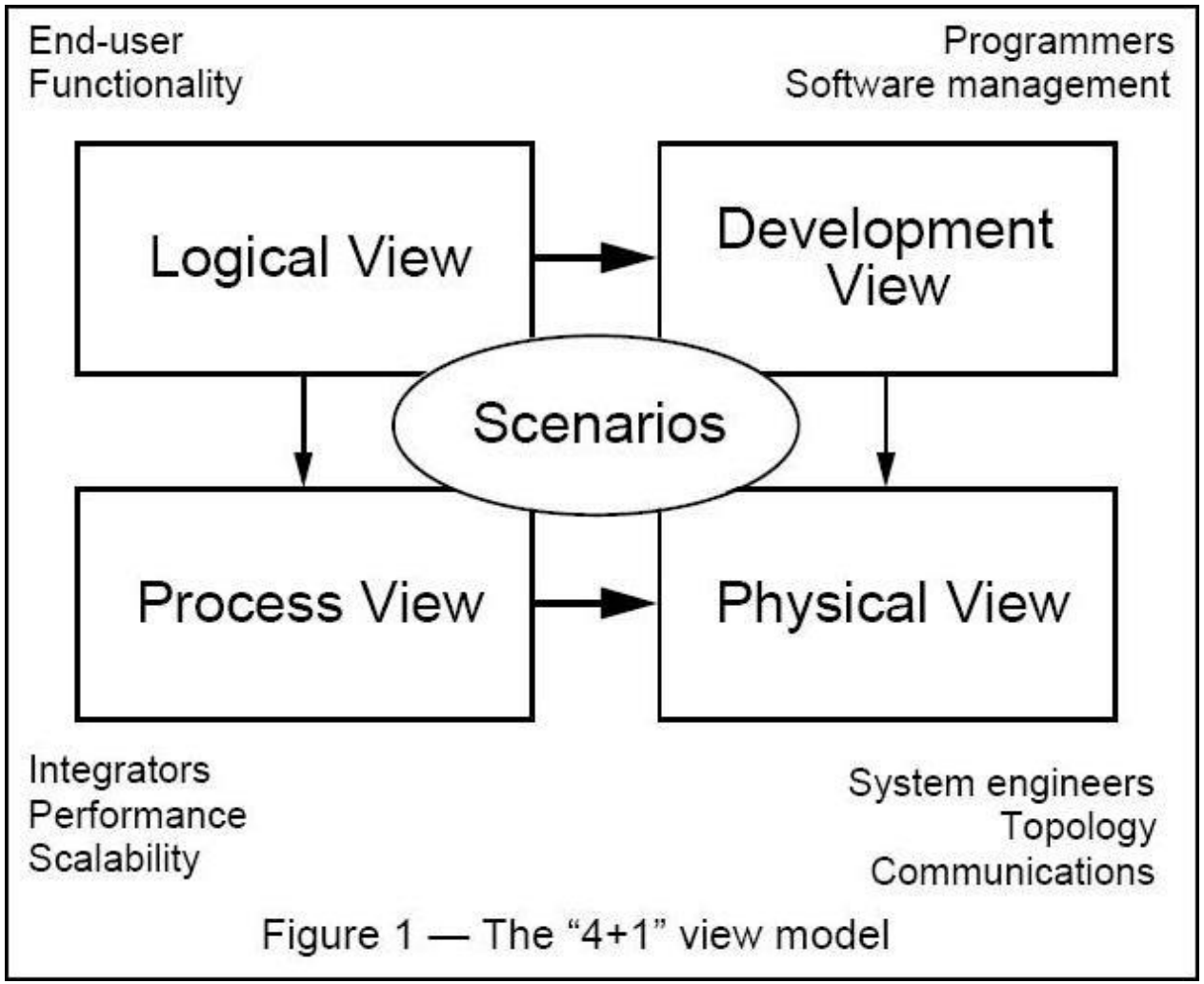


Figure 1 — The “4+1” view model

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

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

Дизайн на архитектурата

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

Архитектурен дизайн

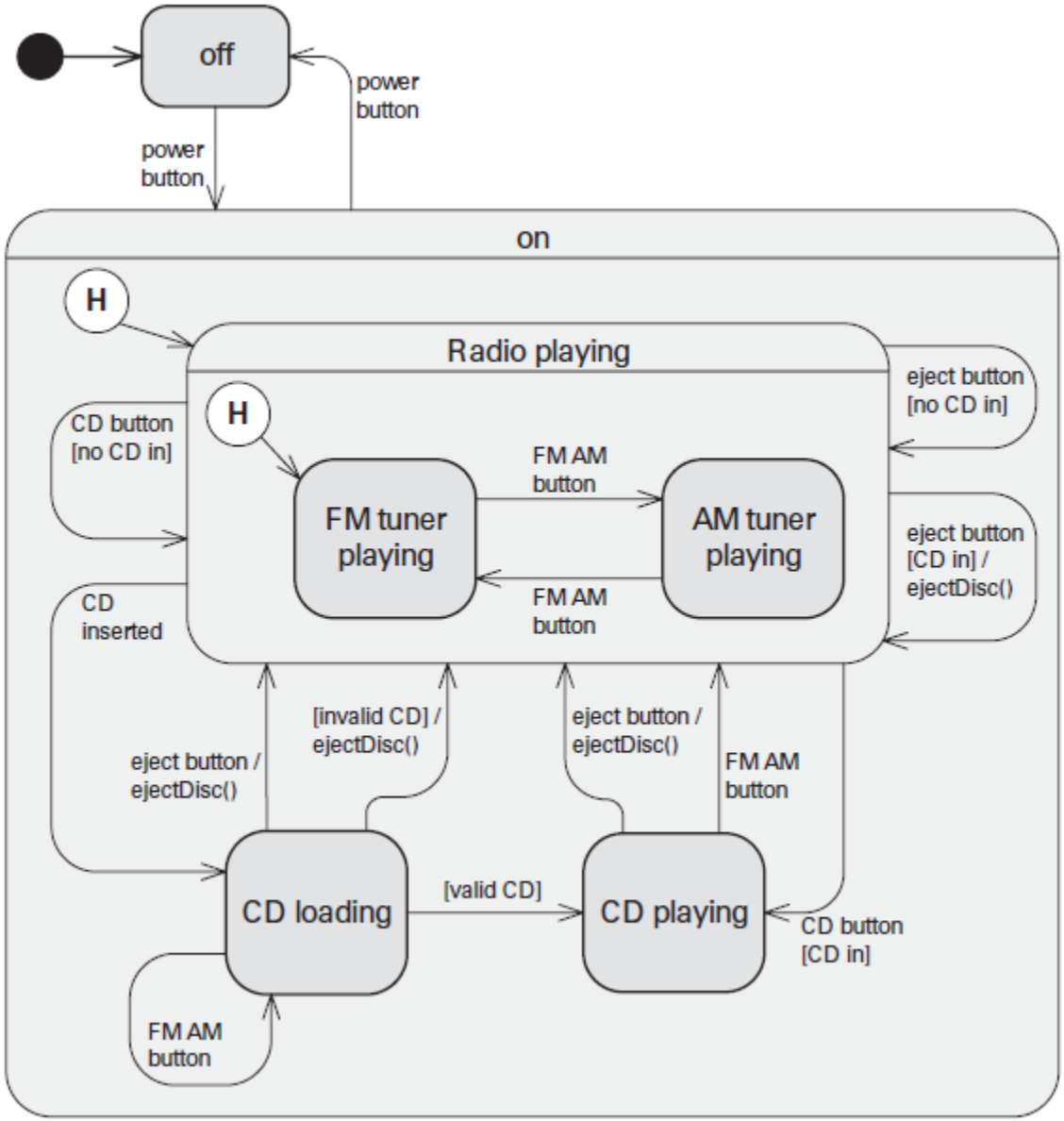
- Процесът на проектиране, при който се идентифицират подсистемите на една система и начина по който те комуникират и биват управлявани в рамките на системата
- Резултатът от процеса на архитектурен дизайн е документация (***описание на архитектурата***)

Описание на архитектурата

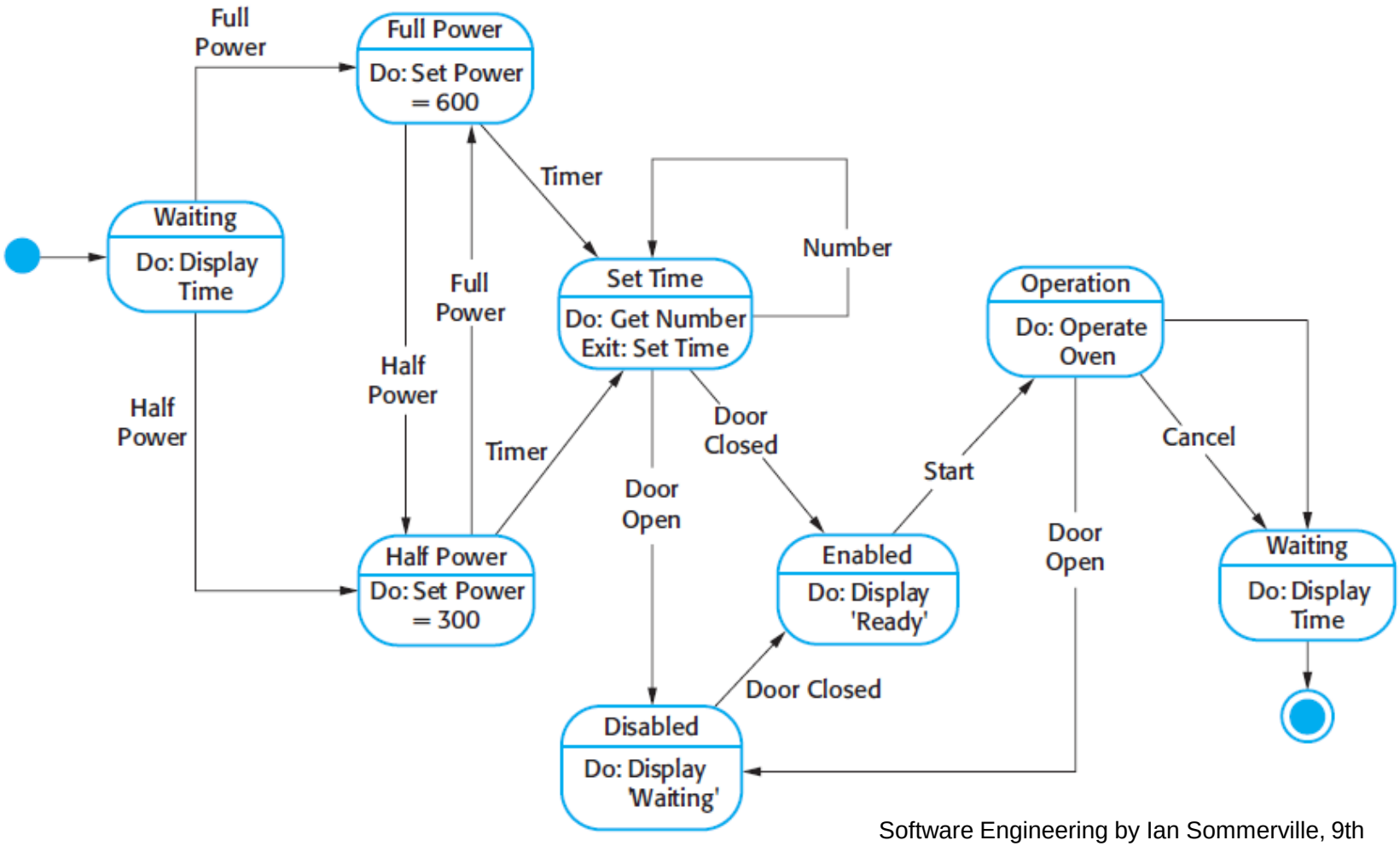
- Документация, модел, описание?
  - Кое от трите?
  - Всички заедно?
- Защо е нужен модел на софтуерната система?

Какво наричаме описание на архитектурата

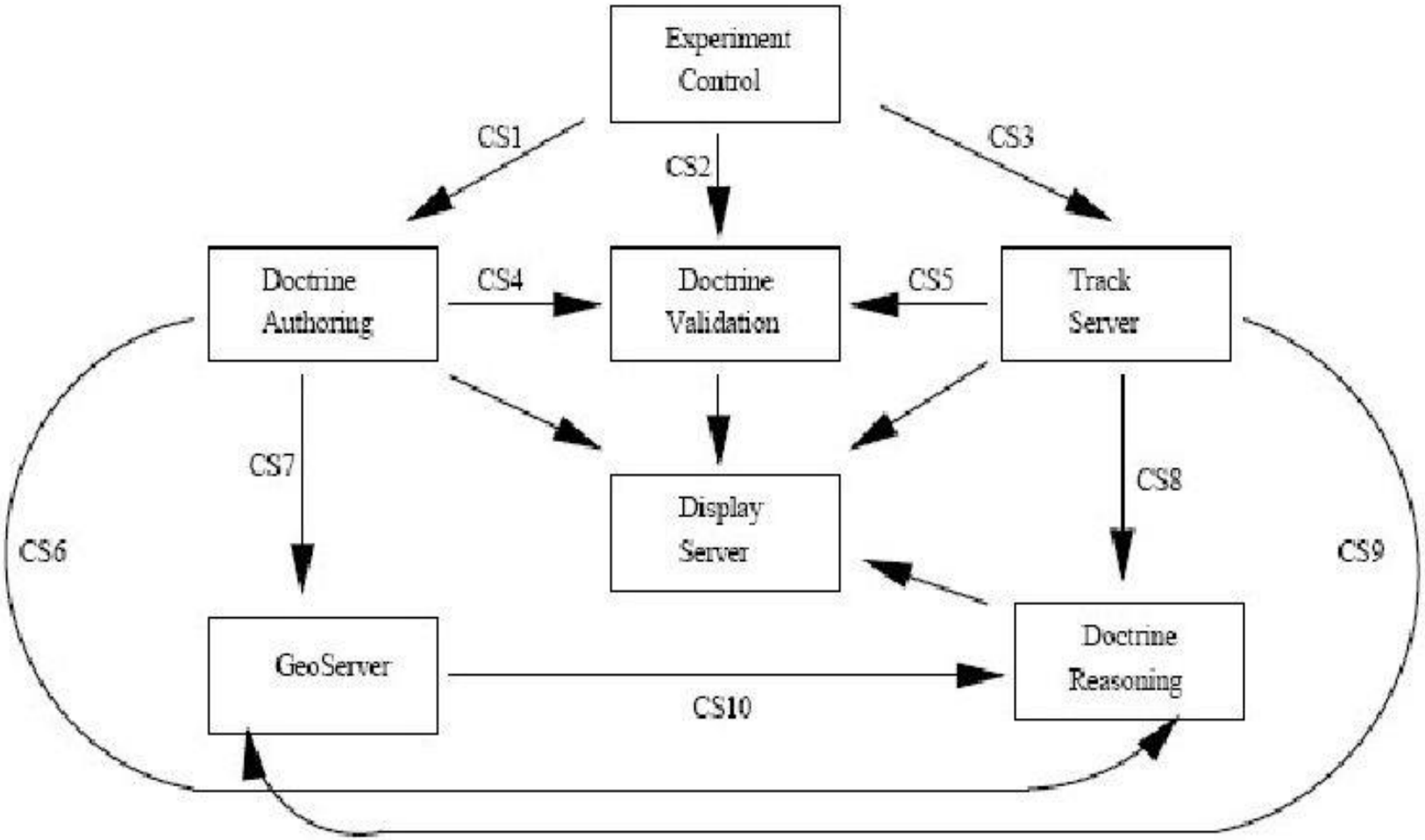
- Описание от тип “кутии и линии
  - Твърде абстрактно – не показват нито последователността на взаимодействието между компонентите, нито дават повече информация за зависимостта между подсистемите (и компонентите) и т.н.
  - Въпреки този недостатък, този тип описания са полезни за комуникация между участващите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

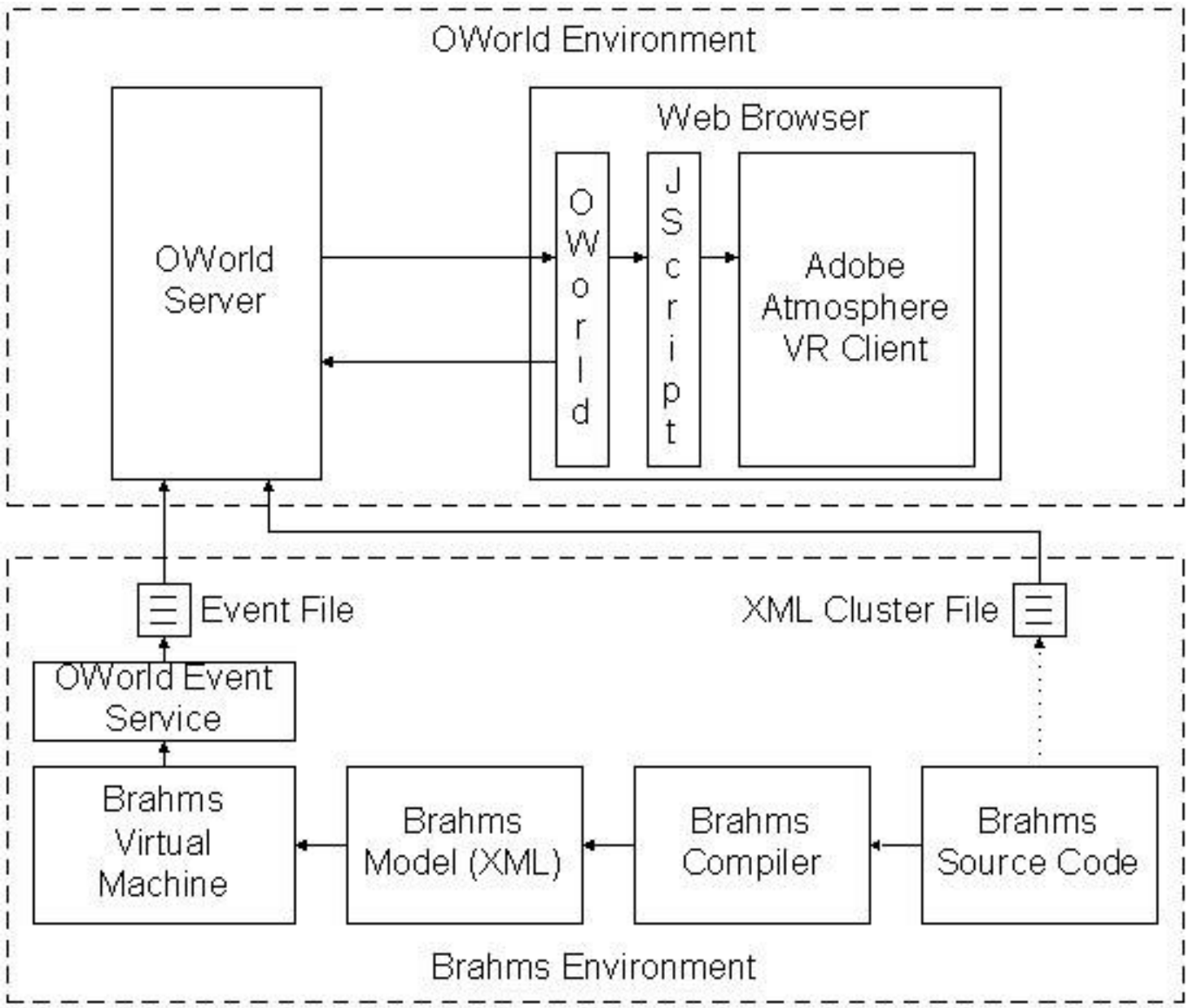


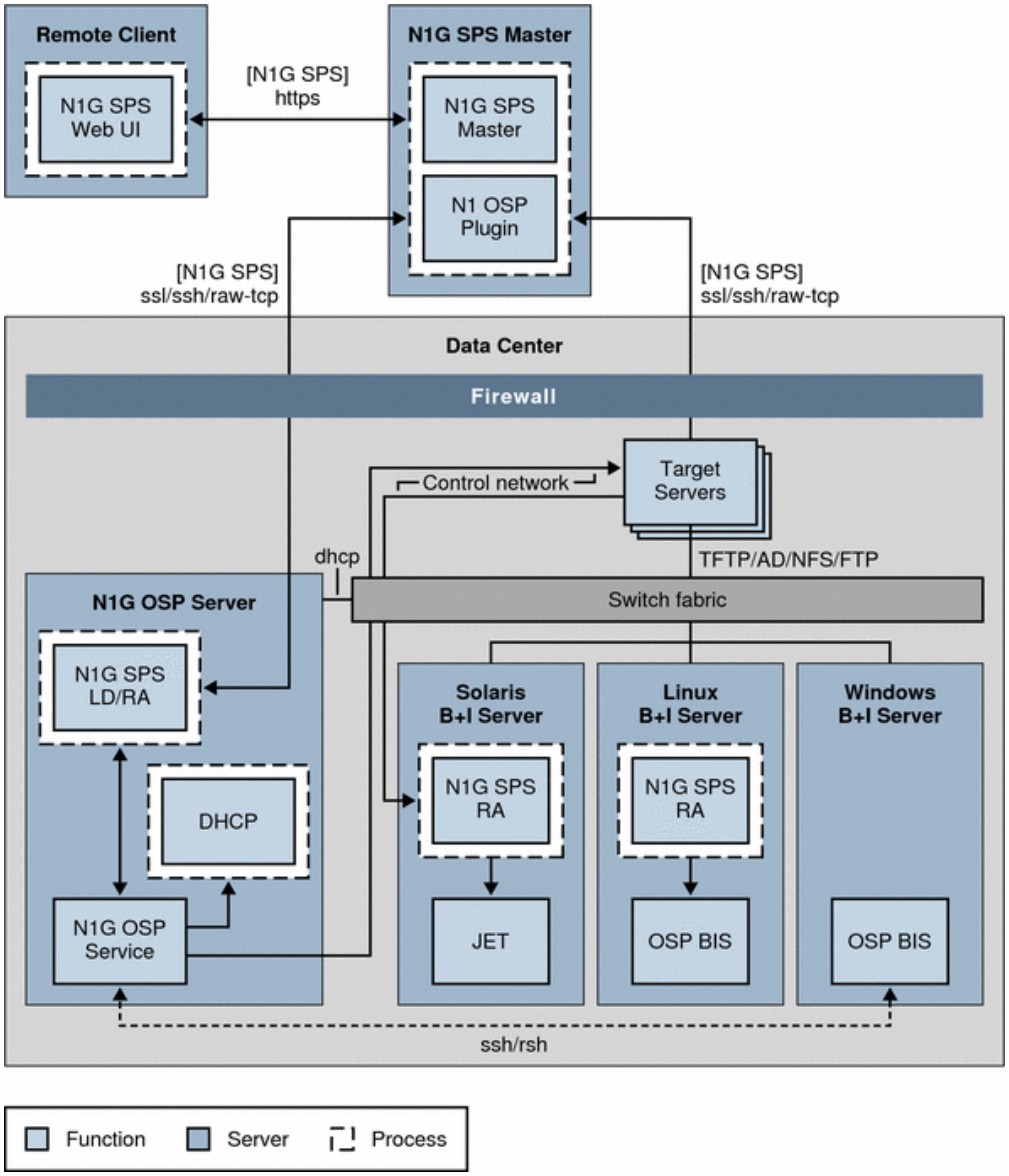
Paul Clements et all., Documenting Software Architectures: Views and Beyond, Pearson Education, 2011



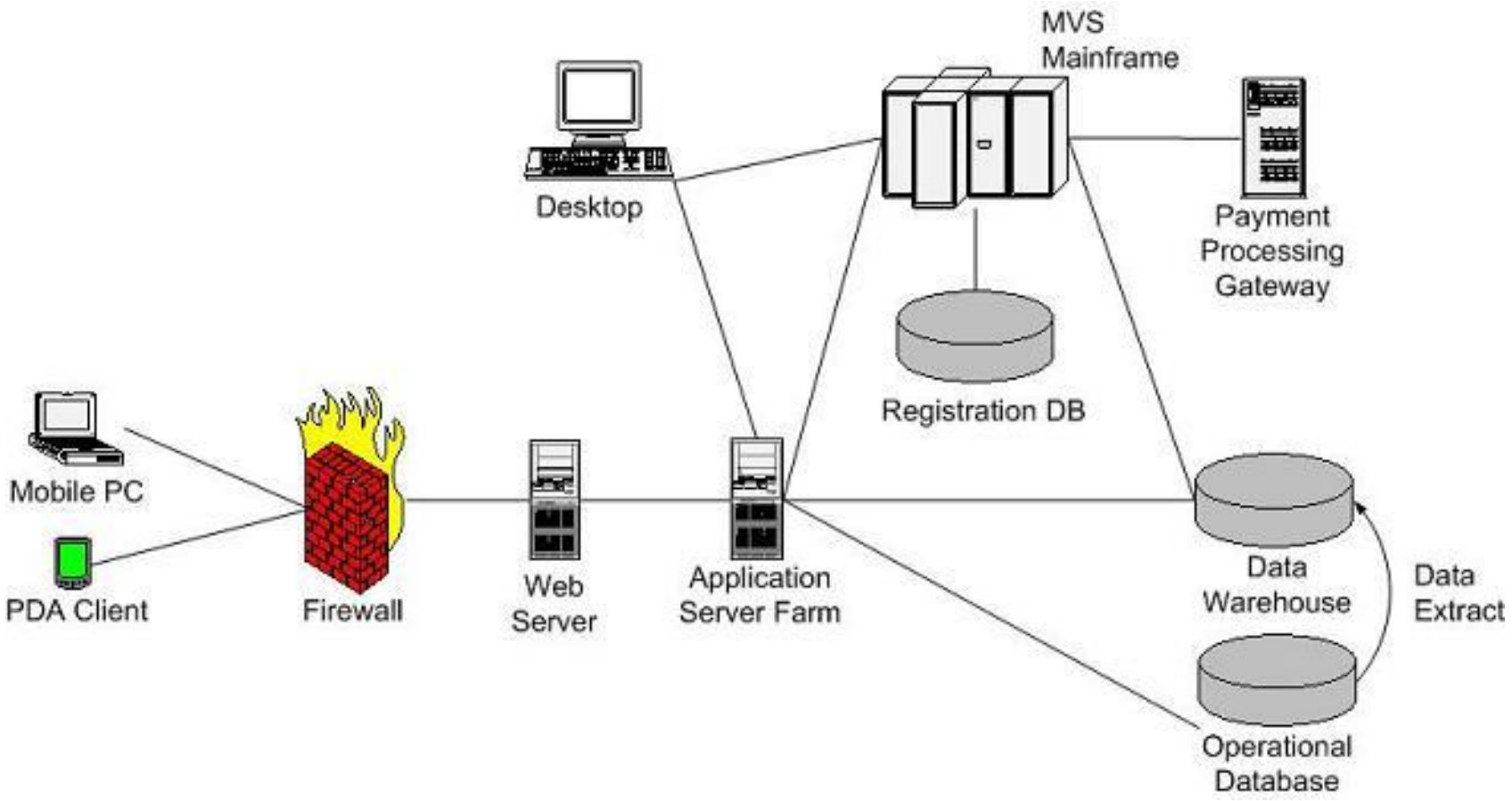
Software Engineering by Ian Sommerville, 9th edition (2010), Addison-Wesley Pub Co;


Experiment Control
CS1
CS2
CS3
Doctrine Authoring
CS4
Doctrine Validation
CS5
Track Server
CS7
CS8
CS6
Display Server
CS9
GeoServer
Doctrine Reasoning
CS10


OWorld Environment
OWorld Server
Web Browser
OWorld
JScript
Adobe Atmosphere VR Client
Event File
XML Cluster File
OWorld Event Service
Brahms Virtual Machine
Brahms Model (XML)
Brahms Compiler
Brahms Source Code
Brahms Environment


Remote Client
N1G SPS Web UI
[N1G SPS] https
N1G SPS Master
N1G SPS Master
N1 OSP Plugin
[N1G SPS] ssl/ssh/raw-tcp
[N1G SPS] ssl/ssh/raw-tcp
Data Center
Firewall
Target Servers
Control network
dhcp
TFTP/AD/NFS/FTP
Switch fabric
N1G OSP Server
N1G SPS LD/RA
DHCP
N1G OSP Service
Solaris B+I Server
N1G SPS RA
JET
Linux B+I Server
N1G SPS RA
OSP BIS
Windows B+I Server
OSP BIS
ssh/rsh
Function
Server
Process

Физически изглед



http://www.agilemodeling.com

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 документира 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

- Нека проектираме клас „човек“, за два вида софтуерни системи
  - Система за управление на ресурсите в университет
  - Ситстема 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

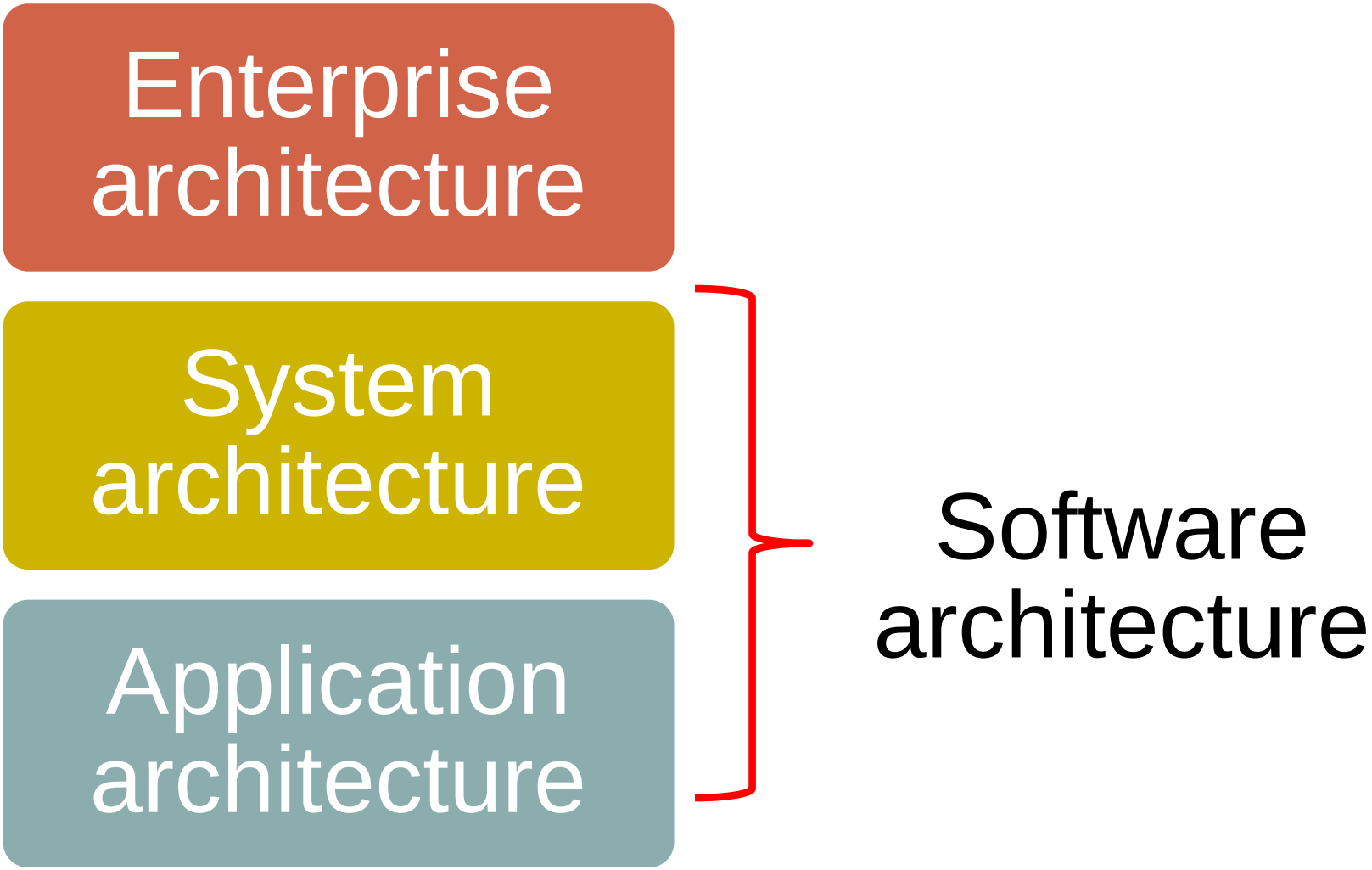
Всички системи ли имат архитектура?

- **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

По-общо понятие за архитектура

- Организационна архитектура (Enterprise architecture)
  - Основните процеси, технологичните и бизнес-стратегии в дадена организация
- Системна архитектура (System architecture)
  - Организацията на програмите и инфраструктурата върху която те се изпълняват
- Архитектура на приложението (Application architecture)
  - Организация на приложение, подсистема или компонент

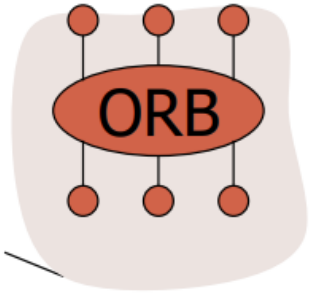
По-общо понятие за архитектура



Седемте нива на софтуерната архитектура*

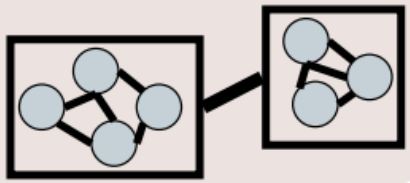
Глобална архитектура

Корпоративна архитектура

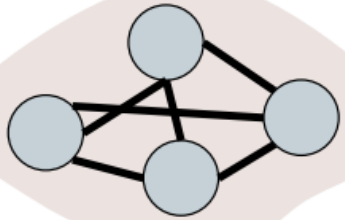


Системна архитектура

Подсистеми



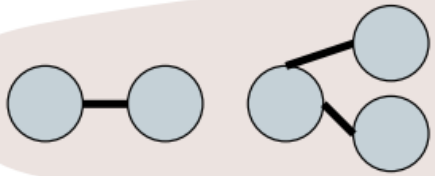
Приложна архитектура



Приложни рамки

Макро-архитектура

шаблони за проектиране



Микро-архитектура

Обекти

ОО програмиране

* Mowbray and Malveau, 1997

Влияние на ЗЛ върху архитектурата

- Тези интереси най-често си противоречат;
- Архитектът е в неблагоприятна позиция – какъвто и ход да предприеме, все някой от списъка със ЗЛ ще е недоволен;
- Ролята му е да балансира между различните ЗЛ за бъдат конкретните интереси отразени в спецификацията на изискванията!

Влияние на организацията върху архитектурата

- Основното влияние идва от целите, заради които се създава системата (изискванията ги отразяват най-пълно);
- Други влияния са:
  - Текущо състояние на организацията;
  - Употреба на предишни разработки;
  - Организационна структура;
  - Стратегия за дългосрочни инвестиции;

Влияние на технологиите

- Частен случай на влиянието на опита и средата на архитекта е влиянието на текущите технологии:
  - Индустриални стандарти;
  - Най-добри практики;
  - Преобладаващи инженерни техники;
- В настоящия момент модерни са уеб-базираните, ориентирани към услуги софтуерни архитектури.

Влияние на опита на архитекта

- Знанията и уменията на архитекта влияят върху създаваната СА:
  - Ако архитектът има положителен опит с даден подход, вероятно ще го използва отново;
  - Обратно, ако резултатите са били катастрофални, най-вероятно ще се въздържи;
  - Подходът ще зависи и от това къде, какво и колко е учил и чел архитекта;
  - Дали се е сблъсквал с успешни/неуспешни подходи и/или реализации;
  - Наклонности за експерименти;

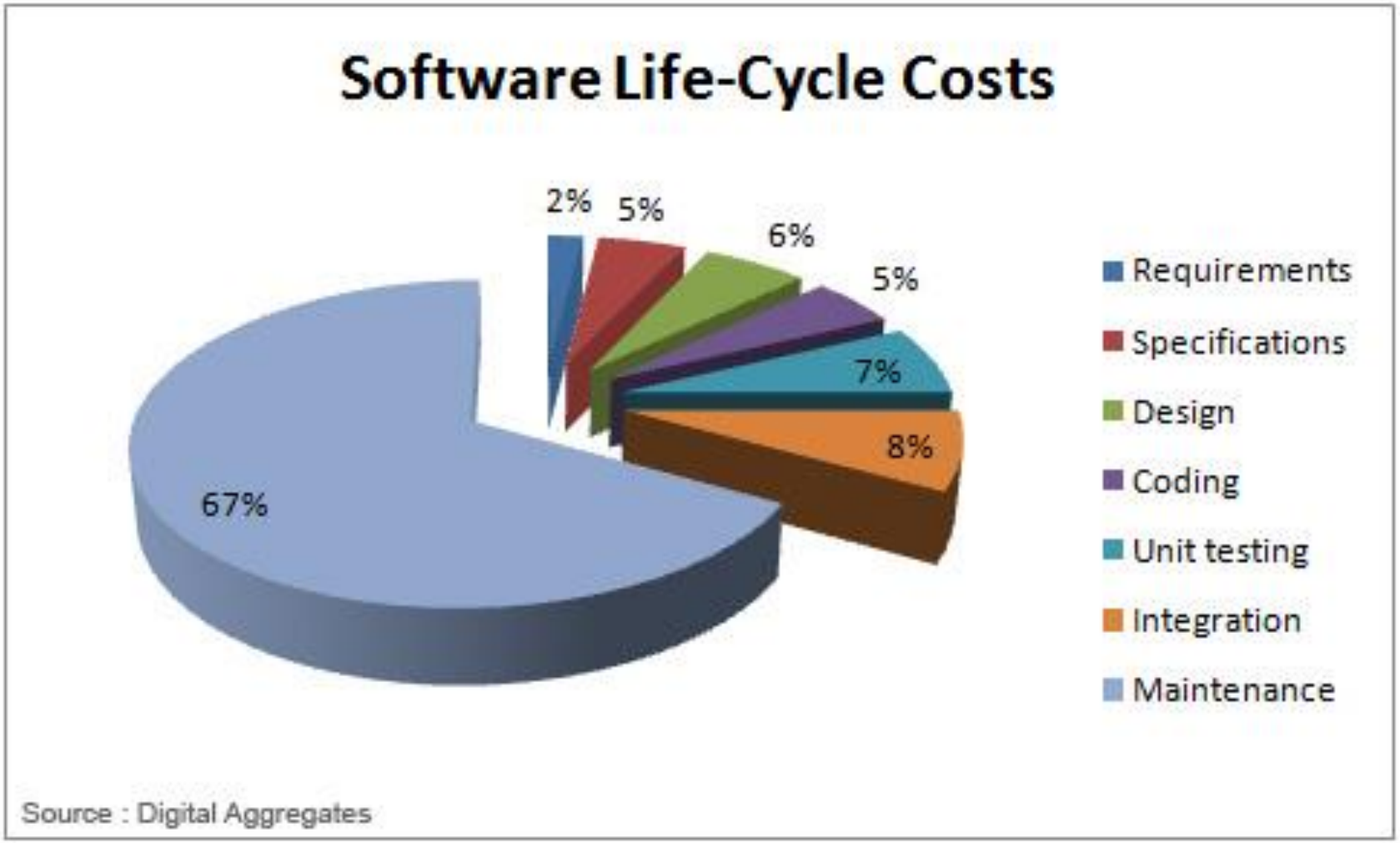
Reverse engineering

- 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



Обобщение

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