**Skills of the Software Engineers**

* Coding skills -20 % (programming Basic, Fundamentals, IDE)
* Algorithmic thinking – 30 % (разбиваме проблемите на по-малки), трябва да решим 1000+ задачи
* Fundamental software development concepts – 25 % (OPP, Functional programing, Asynchronous program, DB, Web technologies (HTTP, JS, DOM, AJAX, REST), Software engineering (source control, agile))
* Programming languages and software technologies -25%
* OOP – Inheritance, Interfaces
* Functional Programing – Purely functional languages, Impure functional languages, Multi-paradigm languages, lambda functions
* Data structures – Arrays, list, stacks, queues, Trees(DFS,BFS),
* Component-based software development – Console functions, Excel functions, Date picker, UI controls, - generally library
* Event-driven programming - Event source, Event handler - чака събитие(examp: натискане на мишката)

**Software Architectures – чертеж или диаграма**

* Monolit apps – цялото приложение в едни APP
* Client - Server model – разделено е на части
* DNS - сървърът, който съхранява информацията за адресите на уебсайтовете и техните съответстващи IP адреси.
* **Front-end (back-end – нужно да знае, независимо че си front-end)**

- Web - HTML + CSS + JavaScript + JS libraries

- Web front – end frameworks – UI(**React (facebook)** Or Angular(google))

- React Native(не е много добре) – за Android и iOS

- Mobile front - end: Android UI, SwiftUI

- Desktop front – end – XAML(Microsoft), UIKit(Apple)

- Dom – document object model(дървото на страницата)

- AJAX – зарежда данни/промени на вече отворена страница, без да я презарежда.(exp. Коментар във Facebook). За AJAX заявките стоят RESTful APIs предоставя данните.(GET, PUT, UPDATE, DELETE).

- Template engines - един и същ шаблон и различна информация(общ template).

- Routing - на базата на пътя(URL) кода да знае какво да зареди.

- Routing lib. – напасват път към изглед

- Graphical User Interface (GUI) – input, buttons, готови компоненти

- Libraries - малко парче което се подпъхва в нашия код

- Frameworks – нашия код се подпъхва в Frameworks

- React (library)

* **Back-end (C#/.NET back - end )**

- ASP.NET MVC, Web API, Entity Framework….

- JavaScript – Node.js, MongoDB

- Data management – виж лекцията за DB

- ORM framework - позволява да пишем на C# а не на SQL(транслира C# в SLQ), Entity Framework e ORM за писане

- MVC(Model View Controller) - позволява да си организираме функционалностите в back – end

\* Controller - bossiness logic, и прави манипулации – позволява да обработим заявки HTTP GET/POST

\* View - визуализира данните – рендерираме HTML +CSS

\* Model – държи данните

\* Web MCV frameworks : ASP.NET MCV

* **Virtualization And Cloud**
* Cloud – плащаме точно толкова колкото използваме.
* Container image – подпъхваш приложението в контейнер, и няма проблеми с версии и хардуер.
* Docker – платформа за Containers, писане само за команди, няма код
* **Operating Systems** – трябва да имаме базови познания по Linux, Windows - инсталация, конфигурация , file systems, users …, работа с cmd, power Shell,
* Software Engineering
* SCS(Source control systems) – github
* QA
* Testing
* Code Reviews
* Issue Tracker
* Unit test - тест за определено парче код
* Project Tracker
* GitHub има вграден
* **3-tier and multi-tier architecture**
* Presentation trail – това което се вижда
* Business logic trail - обработка на данните
* Data management trail - DB
* SOA and microservice

**Offtopick**

* Сайтове за довършване на код - Online code challenges, hacker ran, top coder
* Игра на Кенов : Flappy Bird с Unity 3D
* Kalin Tsekov Git Hub - виж го!!!
* YouTube – Math for Game Developers