5. Проектиране

5.1 Архитектура

Софтуерната архитектура е съвкупност от важни решения за организацията на програмните системи. Тя се определя от високо ниво структури от софтуерни системи, дисциплината от създаването на такива структури и документация на тези структури. Тези структури са необходими за аргументиране на софтуерните системи. Всяка структура съставлява софтуерни елементи, връзки между тях и свойствата на двете - елементи и връзки. Архитектурата на софтуера е метафора, аналогично е на [архитектура](https://bg.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) на сграда.

Чрез Софтуерната архитектура се вземат структурни решения, които са скъпи за промяна след реализация. Софтуерната архитектура включва специфични структурни варианти от възможности в дизайна на софтуера. Например, системите които контролират космически совалки които трябва да са много бъзи и много надеждни. Следователно, необходимо е да се избере подходящият компютърен език. В допълнение, за да задоволи нуждите за надеждност би могло да се избере на-добрият вариант, да има множество излишни и независими произведени копия на програмата и да се задействат тези копия, докато тече многократна проверка на резултатите.

Документирането на софтуерната архитектура улеснява комуникацията между заинтересованите страни, позволява ранни решения за проектиране на високо равнище и повторно използване на проектните компоненти.

Характеристики на една софтуерна архитектура са:

* Архитектурата спомага за използваемостта. Тя дава възможност на потребителя да поема инициативи например, да прекрати дълго изпълняващи се операции или да се откаже от последната команда;
* Системата трябва да може да придвижва реакцията на потребителя;
* Системата трябва да дава възможност на потребителя да бъде ефективен в работата си чрез прилагане на тактики;
* Нуждата от софтуерната архитектура е предизвикана от необходимостта за солидна основа при големите и сложни системи. Тя е гаранцията за дълъг живот на системата;
* Ако не са предвидени различните сценарии и начини на използване на системата, общите проблеми и начините на справяне с тях, целите в дългосрочен план, то тогава софтуерният продукт е в риск. А това струва време и пари! Разбиране на системата от всяко заинтересовано лице, нейното структуриране на елементите е друга необходимост;
* Софтуерната архитектура играе ролята на инструмент за комуникация, обосновка при вземане на решения, средство за анализ и развитие на системата;

Софтуерната архитектура е един вид“интелектуално разбираема“ абстракция на сложна система. Тази [абстракция](https://bg.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%86%D0%B8%D1%8F) осигурява редица предимства :

* Дава основа за анализ на поведението на софтуерната система, преди тя е да е била построена. Способността да се провери, че една бъдеща софтуерна система отговаря на нуждите на своите заинтересовани страни без тя да е построена още, представлява значително намаляване на разходите и рисковете.
* Тя осигурява основа за повторно използване на елементи и решения. Пълна софтуерна архитектура или част от него, може повторно да се използва в няколко системи, чиито участници изискват подобни характеристики и функционалности. Това спестява разходи за проектиране и намалява риска от проектантски грешки.
* Тя подкрепя ранните дизайнерски решения, които влияят на разработката, разгръщането и живота на поддръжката. Вземането на ранни решения предотвратява преразходи в бюджета.
* Улеснява комуникацията на заинтересованите страни, допринася за една по-добра система отговаряща на нуждите им.
* Софтуерната архитектура спомага управлението на [риска](https://bg.wikipedia.org/wiki/%D0%A0%D0%B8%D1%81%D0%BA). Тя помага за намаляване на шансовете за провал. Дава възможност за намаляване на разходите в сложни IT проекти.

### Архитектурни тактики

<https://bg.wikipedia.org/wiki/%D0%A1%D0%BE%D1%84%D1%82%D1%83%D0%B5%D1%80%D0%BD%D0%B0_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0>