Структури за софтуерна компания и ролята на софтуерните инженери в тях

1. Видове структури
2. хоризонтална – началникът отговаря пряко за всички и всеки поема по възможно най-много длъжности; подходяща е за малки фирми

* предимства:
* гъвкавост, защото достъпът до началника е бърз и защото при проблем или идея от един служител могат да бъдат отразени бързо в няколко направления
* по-голямата отговорност на всички им дава самочувствие и мотивация
* по-бърза отзивчивост към клиента
* бързи резултати
* не е от голямо значение качествата на управителя
* недостатъци:
* всички са шефове, от което възникват концептуални противоречия, липса на координация в действията и разминавания в резултатите
* високо натоварване на ръководителите и най-вече на началника
* лошо качество на резултатите

1. вертикална – подробно разпределение на ролите чрез супервайзъри

* предимства:
* по-ясен мениджмънт и по-ясни отговорности
* по-прецизно проследяване на дейностите от по-високо поставените ръководители
* по-добър контрол над действията
* по-ефективно разрешаване на локални проблеми
* по-качествени резултати
* ясна йерархия и по-висока мотивация при възможност за повишение
* чувство за сигурност
* ниска възможност за вземане на грешни решения и предприемане на грешни действия
* недостатъци:
* тромавост: нужда от повече време за вземане на важни решения, поради „пътя“ до високопоставените управители
* условия за ниско самочувствие на нискостоящите, което е предпоставка за слаба мотивация и ниска ефективност
* силна зависимост от качествата на управителя
* възможност за липса на компетентност от супервайзъра, ако се съди на база неговия опит, а не на изискванията към него и възможностите му

1. Основни принципи при разработка на софтуер
2. познаване нуждите на клиентите, за които е предназначен софтуера
3. познаване недостатъците на средата им и/или проблемите им
4. предоставяне решения на проблемите им
5. предоставяне на възможности
6. Обща фирмена структура
7. производствен отдел (production department) – от тук се определя минималната себестойност на продукта

* разработка на софтуер
* инженери по изграждане на бизнес логиката
* инженери по бази данни
* дизайнери и UX експерти
* инженери по изграждане на UI-а
* инженери по осигуряване на качеството
* бета тест инженери
* project мениджър
* управление на риска
* следене на графика и развитието
* production мениджър
* проучвания и анализ
* продуктов мениджър
* бизнес аналист
* превежда бизнес нуждите към продукта на разработчиците

1. маркетинг отдел

* ценообразуване
* perceived value (стойността, на която клиента оценява продукта)
* реклама
* PR
* брандинг
* силно обвързан с търговския отдел

1. вътрешно-организационни отдели и мениджъри

* facilities manager – поддържа работната среда (бюра и пр.)
* правен консултант
* счетоводители
* ИТ мениджър
* security мениджър
* системен администратор

1. Управленски позиции (C – chief, 2nd letter – environment, O – office (CEO, CMO))
2. Stockholders, shareholders, IPOs – хора, които притежават дялове и/или акции на компанията и формират борд на директорите
3. Stakeholders – всички хора, които са зависими от компанията (клиенти, служители, доставчици и т.н.)
4. Организиране процеса на разработка
5. изисквания на продукта
6. дизайн

* архитектура
* UI

1. Разработка

* Технологии – имаме ли възможност да използваме дадена технология
* софтуер
* хардуер

1. осигуряване на качеството
2. обратна връзка от резултатите и от отзиви
3. Модели за разработка на софтуера
4. Waterfall – праволинейно разработване

* фази
* събиране на изисквания (документация)
* организиране на ресурсите
* анализ
* дизайн (архитектура)
* производство
* тестване
* пакетиране
* поддръжка
* предимства
* подходящ за клиенти, които имат строго конкретни изисквания и очаквания
* оправяне на грешки още при дизайна
* изпълнителност
* документиране
* яснота
* подходящ за стабилни проекти
* недостатъци
* бавна обратна връзка на клиента
* бавна разработка, от което може да се променят нуждите на клиента

1. Agile – JIT (just in time)

* фази
* дизайн
* производство
* след което започва паралелно:
* документиране
* тестване
* поддръжка
* предимства
* подходящ за клиенти, които не знаят какво искат
* предразположение се за координация и съвместна работа
* служители с повече умения не стоят да дойде техния етап от процеса
* производството се реализира на равномерни интервали
* възможни промени в процеса на разработка
* бързо реагиране на текущите изисквания на клиентите
* видове
* Kanban – дъска с карти (To do, In work, Done)
* предимства
* опростеност и яснота
* гъвкавост и свобода на действие
* лесно за проследяване
* по-лесно планиране
* по-малко напрежение от крайни срокове
* приоритизиране
* недостатъци
* твърде гъвкаво, което е предпоставка за лоша дисциплина
* трудно планиране на срокове
* Scrum – планиране в кръг
* предимства
* подходящ, за по-комплексни случаи, когато има много неизвестни ситуации
* бърза комуникация
* паралелно разработване на функционалности
* възможност всеки да реализира идеи
* силно изразена ясно, заради срещата с клиента при всеки произведен sprint
* по-лесно прогнозиране на крайните срокове
* лесно се изчислява времето за производство
* недостатъци
* труден за реализация в по-големи мащаби
* труден за реализация при не добре сработени екипи
* при ненапълно ясни изисквания, са възможни провалени sprints
* изисква се голям ресурс за планиране, прегледи и ретроспективни анализи