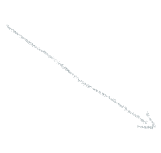
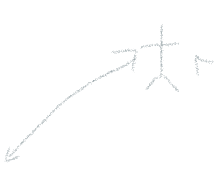
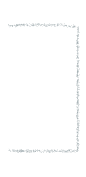
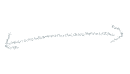
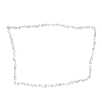
Case-verktøy (computer added software engineering):

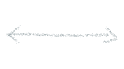
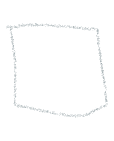
* Upper case verktøy
* Støtter aktiviteter tidlig I prosessen (støtter analyse)
* Lower case verktøy
* Støtter design og realisering (koding)
* I case verktøy
* Støtter alle fasene (Finnes egentlig ikke, kun i teori)

Generell arkitektur:



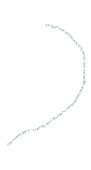
systemutvikler

Upper case repository Lover case

varktøy utviklingsdatabase verktøy

use-case beskrivelse 1

DFD-fragment 1

Deler av DD

ERD

Skjermbildeutkast

Egenutvikling vs. Standardsystemer:

* Standardsystemer er noe en bedrift kan kjøpe, den er ferdig utviklet. Det utviklere ofte tjener penger på her er f.eks. installasjon/ innføring,drift, vedlikehold, tilpasning til bedriften, ol.
* Hvor i prosessen gjør en valget for om man skal utvikle eget system eller kjøpe standardsystem? Mange bommer her ved å velge et system for tidlig i prosessen, altså det er for tidlig å si hva systemet skal gjøre. Kravspesifikasjon må være på plass for å kunne se om systemet sjekker ut alle bokser for hva man trenger.
* Uklare arbeidsforhold mellom maskinvare, programvare og bruk
* To aktiviteter/ faser gjelder ikke:
* Design og realisering erstattes med vurdering og valg

Innføring, drift og vedlikehold:

* Test av applikasjon
* Gjennomgang av kode / syntaks og testing av funksjonalitet/ logikk inngår ikke her; det er vi ferdig med på forhånd
* Alle moduler/ deler testes enkeltvis
* Det viktigste blir å teste systemet i sin helhet/ kjøre gjennom sykluser i programmet
* Testplan
* Teste deler vs. Teste helhet
* Testdata vs. Teste på reelle data (tester man på reelle data, kan det være lurt å teste på kopien av reelle data, just in case)

Systeminnføring:

* Erstatte gammelt system med et nytt system
* Det finnes ulike strategier:
* Direkte overgang (setter sluttdato for gammelt system, og etter den datoen blir det nye systemet tatt i bruk) billigst metode med størst risiko
* Parallellkjøring (kjører gammelt system samtidig som man begynner med det nye systemet i en periode) dyr metode med liten risiko
* Pilotering/ pilot innføring (virksomheten fortsetter å kjøre gammelt system; deler av virksomheten tar i bruk det nye systemet enten som direkte overgang eller parallellkjøring. Når tidsperioden er over, tas det nye systemet i bruk av resten av virksomheten)
* Innfasing
* Uavhengig av strategi:
* Viktig å få med historiske data fra gammelt til nytt system
* Data fra gammelt system må innom en konverteringsalgoritme slik at de havner på rett sted i det nye systemet

Tenk kvalitet, data:

* I gammelt system
* Etter uttak/
* Etter konvertering
* Etter innput i nytt system

Brukeropplæring:

1. Hva slags opplæring skal tilbys?
2. Hvordan skal opplæringa gjennomføres?

* Vi må ha skreddersøm mot arbeidsoppgave (tenke på hvem som trenger hvilken opplæring)
* Fokus på arbeidsutførelse
* Dvs. ikke ”one-size-fits-all”