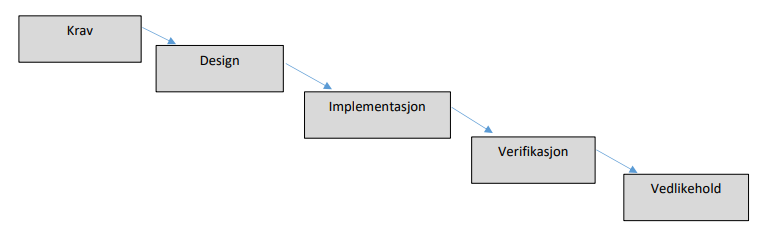
Systemutviklingsmetodikk

Fra 1970 var det vannfallsmodellen som var gjeldende. Derimot rundt år 2000 kom både UP og smidige metoder. Det er smidige metoder som gjelder nå, eksempler er:

* XP
* TDD
* SCRUM
* Kanban
* Domain-Driven Design

**Vannfallsmodellen (ca. 1970)**

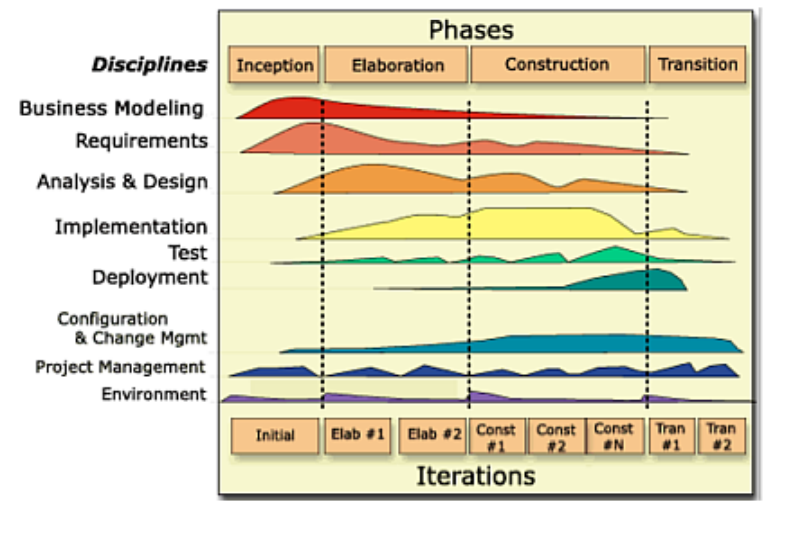


Tung modell siden man går gjennom en og en fase om gangen. Det er forskjellig folk i hver fase, som da krever mye overlevering av dokumentasjon fra fase til fase. Endringer underveis ble derfor vanskelig og dyrt. Mange prosjekter feilet.

**UP (Unified Process)**



UP innførte iterasjoner for å takle endringer underveis. UML-modellering er svært sentralt i UP. Dokumentmalene definerer hvilke modeller som skal brukes. Er use-case-drevet. Vi starter med use-case’r og detaljerer mer og mer med andre typer modeller etter hvert. Bruker faser: inception, elaboration, construction og transition.



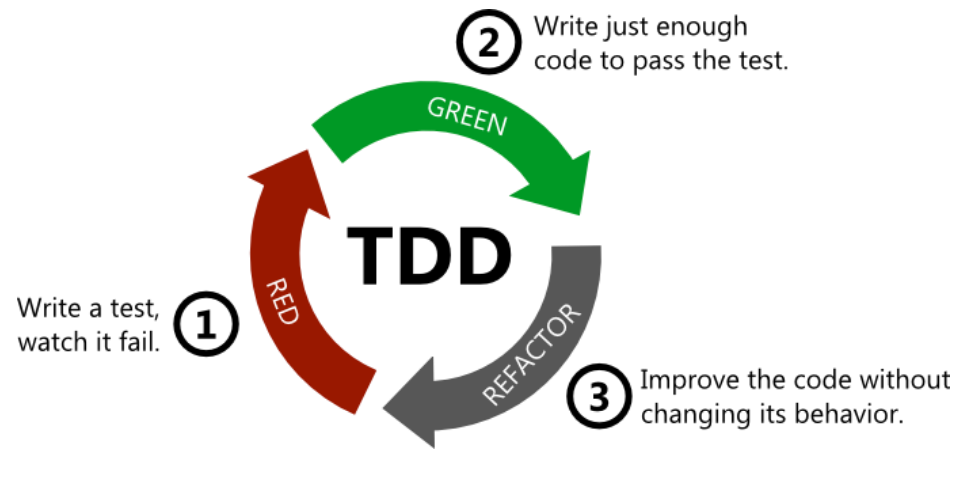
# Smidig utvikling

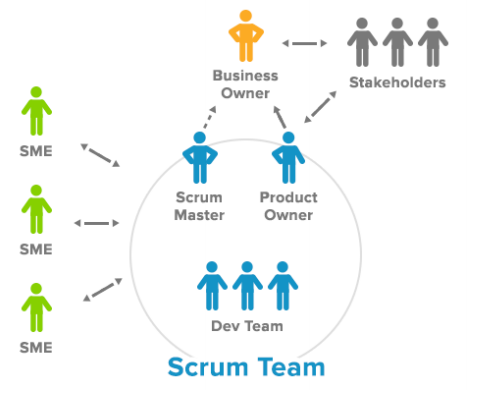
Agile = smidig, rask. Smidig utvikling oppsto på 1990-tallet, men 2001 er likevel et merkeår siden «The Agile Minifesto» ble publisert da. Finnes flere ulike metoder innen smidig utvikling:

**Extreme Programming**

Kanskje den første smidige metoden. Nesten ingen dokumentasjon og modellering. Har 29 regler: Programmer alltid parvis, lag alltid en enhetstest for det du skal lage før du skriver koden, osv...

**TDD (Test-Driven Development)**

Videreføring av XP, men ikke like rigid. Fokus på *enhetstesting*, *akseptanstesting* og refaktorering. 

**SCRUM**

En smidig prosessmetode som ikke forteller oss noe om hvordan vi skal utvikle koden eller modellere krav, men hvordan vi skal jobbe sammen og få fremdrift.

***Team*** på 7-8 personer. ***Stand-up*** møter hver morgen 15 min. ***Sprinter*** (iterasjoner, en, to eller tre uker, med klare mål for hver sprint og demo når hver sprint er ferdig).

**LEAN**

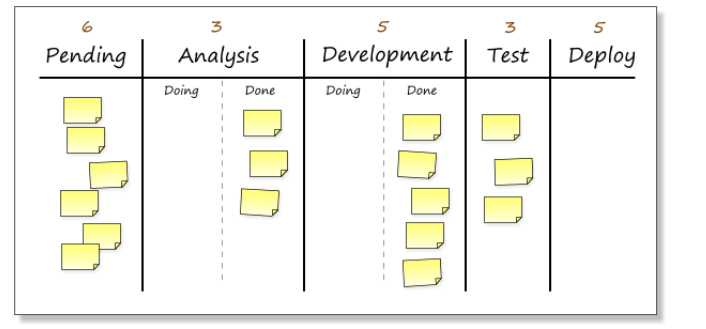
Begrep/tankegang/metode på samme måte som **agil**. Har veldig mange likheter, men kommer fra «Lean manufacturing» og Toyotas produksjonssystem. Lean betyr **slank**, mens agil betyr **smidig**. Men begge metodene er både **slank** og **smidig**, derfor kan det være vanskelig å se forskjellen.

Lean legger hovedvekt på å unngå å gjøre/lage ting uten verdi, eller «waste».

*Prinsipper i Lean Development.*

* **Eliminate Waste**: Alt som produseres må ha en verdi. Unngå funksjonalitet man tror kan bli kjekk senere, men kunden egentlig ikke trenger, dokumentasjon ingen leser, for mye byråkrati osv.
* **Build quality in**: Ha fokus på kvalitet på kode og systemet som helhet.
* **Create knowledge**: Kort feedback-loop mellom kunde(sluttbruker) og utvikler slik at flere ting kan testes og man kan gjøre endringer raskt.
* **Defer commitment**: Vent med irreversible avgjørelser så lenge som mulig, til vi har lært mest mulig.
* **Deliver fast**: Tenk enkelt, unngå over-engineering og få ting ut i produksjon så tidlig som mulig.
* **Respect people**: La utviklerne få ansvar. Blir man redusert til «code monkey» miser man motivasjon og man er ikke effektiv. Ledere bør motivere folk i stedet for å kontrollere dem.
* **Optimize the whole**: For å lykkes må man forstå helheten i et system.

**Kanban**

Stammer fra lean-verdenen. En enkel metode for å unngå flaskehalser i produksjon eller ved utvikling av it-systemer. Man bruker et Kanban-board for å holde styr på oppgaver som er gjort eller som må gjøres i ulike områder.

Tenk at du jobber på et samlebånd med flere stasjoner. Når du mangler oppgaver på din stasjon springer du bort og for å hjelpe andre slik at produksjonen aldri stopper opp.

**Under prosjektet vår 2019**

