## Mål & Syfte

Att med agila metoder skapa en fungerande prototyp enligt kundens beskrivning med hänsyn till iterativa processer.

Målet är att

* kunna identifiera sin egna kapacitet och anpassa arbetet därefter.
* identifiera problematiska mål och bryta ner dessa till mindre, hanterbara mål.
* få färdigheter i att bedöma andras kod enligt bestämda konventioner.

Obs., individuella inlämningsuppgifter publiceras i början av sista veckan i kursen på LMS:en som då är baserat på ovanstående mål. Om du är delaktig i projektarbetet och har tagit del av kursen så ska den individuella uppgiften mer likna en journal över samarbetet och en kortare sammanfattning av kursens innehåll.

## Grupper

Grupper skapas under lektion samma dag som uppgiften publiceras

Om du inte har en grupp, kontakta ansvarig lärare snarast!

Om en gruppmedlem inte går att nå eller är delaktig i arbetet, kontakt då ansvarig lärare så snart som detta uppenbaras, det vill säga **inte** i slutet av arbetet.

## Avgränsningar

Syftet med denna uppgift är att köra fast i vanliga problem som kan uppstå då man samarbetar i grupp, mot denna bakgrund finns det alltså inga “officiella” avgränsningar eller riktlinjer för *hur* uppgiften måste lösas. Gruppen själv väljer de verktyg och teknologier som de vill lösa uppgiften med.

## Checklista

Skapa **Trello** board och kontrollera/skapa följande

* + (i trello) Skapa kolumnerna:  
    “product backlog” -> “ready” -> “developing” -> “code review” -> “done”
* Samtliga deltagare har tillgång till trello
* Bestäm en WIP (work-in-progress) för utvecklings kolumnen
  + Bestäm kapaciteten för gruppen, hur många user stories kan utvecklas samtidigt?

För enkelhetens skull så använder vi ett antal user stories snarare än tid/komplexitet av story:n.

Välj **agil** strategi Scrum med lite Kanban

* Vilka branch:es ska finnas? Github flow [Understanding GitHub Flow and Git Flow | by Yan Min Thwin | Medium](https://medium.com/@yanminthwin/understanding-github-flow-and-git-flow-957bc6e12220)
* Hur ofta görs en commit? Flera gånger per dag
* När ska en branch merge:as till main? Färdigställd user story. (Efter Pull request och code review)
* Hur sker en code review? Tips, använd [pull requests](https://support.atlassian.com/bitbucket-cloud/docs/use-pull-requests-for-code-review/), tilldela en “revieweer” Pull request, använd reviewer. Gör review tillsammans
  + När och hur sker en code review? Tips läs [5 best practises (Atlassian)](https://www.atlassian.com/blog/add-ons/code-review-best-practices)
  + Gruppen bör komma överens om några grundläggande regler för hur koden ska skrivas för att göra code reviews enklare. Till exempel:
* Hur ni namnger variabler och funktioner Beskrivande namn på engelska. Camel case i JS. Harmonisering.
* Hur ni strukturerar er kod Följa kodspråkets struktur och konventioner. Effektiv kod, förebygg upprepade kodavsnitt. Kommentarer för tydlighet vid behov.
* Hur ni kommenterar koden Så lite som möjligt, använd beskrivande namn på variabler och funktioner så gott det går

Det viktiga är inte exakt vilka regler ni väljer, utan att ni dokumenterar era överenskommelser och följer dem konsekvent i projektet.

Planera **gruppträffar**

* 1 träff per dag, rekommenderat med två
  + Under dessa dagar förväntas det hållas minst ett möte för att dela med sig av upptäcker m.m.
* Vad är agendan för respektive möte?
  + Vilka möten är för code reviews?
  + Vilka möten är för att diskutera mål?
  + Vilka möten är retrospektiva?

Dessa möten bör vara korta och fokusera endast på vad som ska bli klart just den dagen.

## Första kundmöte

Kunden har en revolutionerande idé för spelet hänga gubben där respektive spelare har sin egna gubbe. Målet i spelet är att gissa rätt på motståndarens ord för att hänga dennes gubbe, samtidigt som man försöker undvika att ens egen gubbe blir hängd. Den spelare som först lyckas hänga motståndarens gubbe vinner spelet.

Ert uppdrag är att samla på er det som ni behöver för att börja skapa en fungerande prototyp. Inför första mötet är det bra att fundera på vad en traditionell hänga gubbe har för komponenter och vad som kunden eventuellt skulle vilja lägga till för att särskilja detta spel från nuvarande hänga gubbe.

Observera att kunden inte vet vilka tekniska delar som är svåra att utveckla eller som behövs i olika delar av utvecklingen så fundera noga på vilka frågor som är viktiga för att utveckla en produkt enligt kundens vision.

För att sammanfatta →

* Vad behöver gruppen få för information från kunden för att kunna utveckla mjukvaran?

## Redovisning

Då kunden inte ännu har valt hur den vill att programvaran installeras så kommer metoden för redovisning dyka upp den andra veckan i arbetet. Grupperna presenterar slutprodukt den **28:e januari 09:00**, vid komplettering från kunden lämnas arbetet in den **30:e januari 09:00**.