

Project kickoff

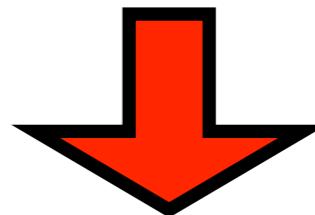
Module 9

Lecture

Operating systems and process oriented programming 2020

1DT096

Course timeline



Mål med
kursen

Mål med kursen (1)

Från kursplanen för kursen Operativsystem och processorienterad programmering (1DT096).

Efter godkänd kurs ska studenten kunna:

- **redogöra** för hur **operativsystem** och **runtimesystem växelverkar** med **maskin-** och **programvara**.
- **förfklara** och **använda algoritmer** och **tekniker** för **schemaläggning** och **synkronisering** i olika system.
- **redogöra** för hur **synkroniseringstekniker** kan användas för att hantera samtidighet i datorsystem, och **bedöma** deras lämplighet i olika situationer.
- **redogöra** för principerna för olika **programmeringsmodeller** av flerkärniga system, till exempel processer, trådar, meddelandeöverföring och "software transactional memory", och **konstruera program** som använder dessa.

Mål med kursen (2)

Från kursplanen för kursen Operativsystem och processorienterad programmering (1DT096).

Efter godkänd kurs ska studenten kunna:

- **använda** verktyg för **versionshantering**, **felsökning** och **testning**.
- **genomföra** ett **strukturerat projektarbete** tillsammans med andra studenter samt, under processen, **reflektera** över och utveckla gruppens samarbete.
- **planera** ett **projekt**, inklusive användande av samarbetsverktyg, så att det kan genomföras inom givna ramar.
- **presentera och diskutera** **kursens innehåll muntligt** och **skriftligt** med för utbildningsnivån lämplig färdighet.

Kурсенс upplägg

Varför är kursen upplagd som den är?

Studenten och arbetsgivaren

I grova drag, vad händer när nyexaminerade studenter möter sin första arbetsgivare?

Studenten

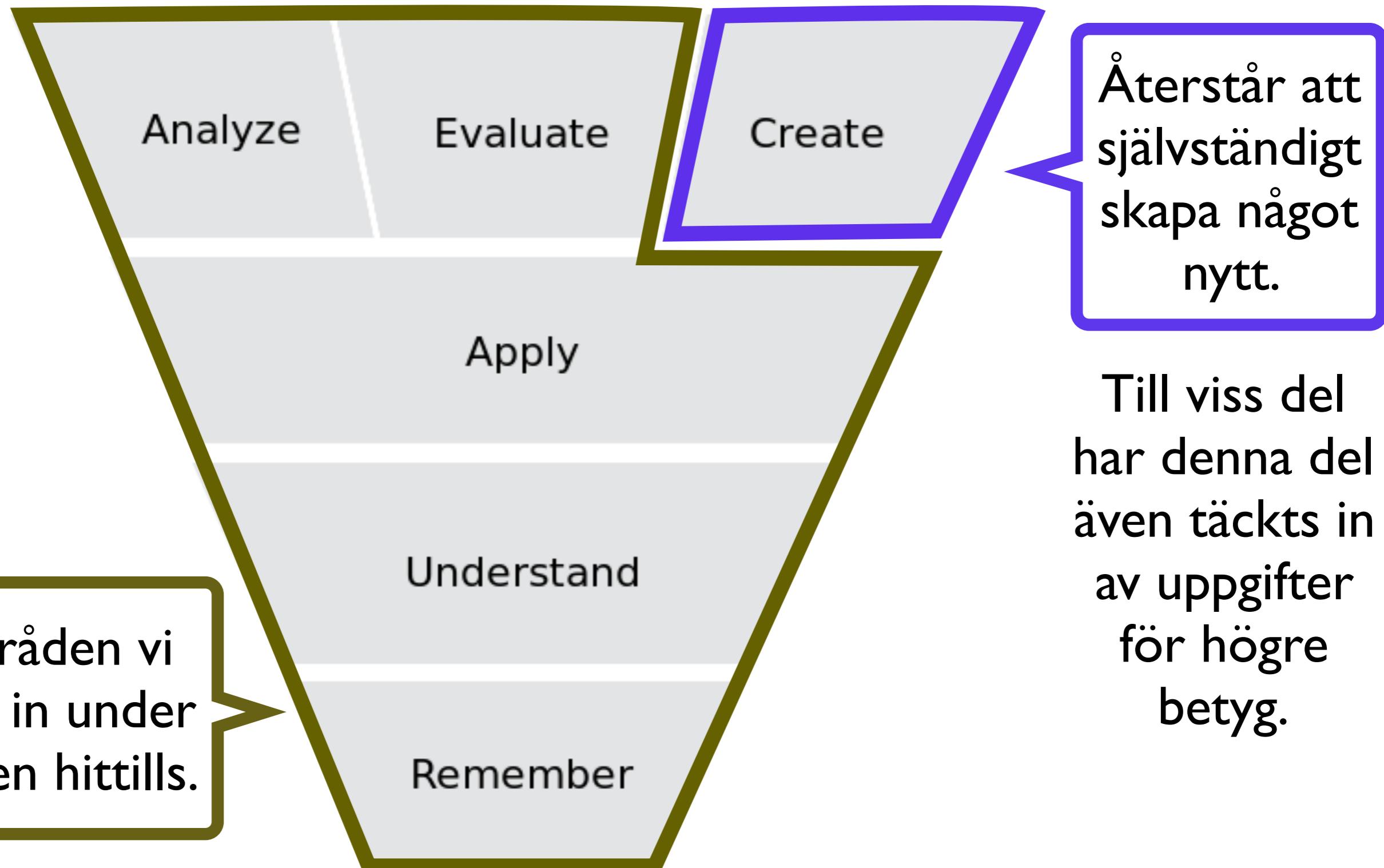
Jag kan väldigt mycket,
men jag får inte
tillräckligt väl
definierade
arbetsuppgifter.

Arbetsgivaren

Studenten kan väldigt
mycket, men klarar inte
av att arbeta
självständigt.

Varför ser det ut så här och vad kan vi göra för att minska gapet?

Categories in the cognitive domain of Bloom's Taxonomy



Open-ended

: not rigorously fixed: as

a : adaptable to the developing
needs of a situation

b : permitting or designed to
permit spontaneous and
unguided responses

open-ended group project

You will not be given a project specification by the teaching staff.

Project proposal

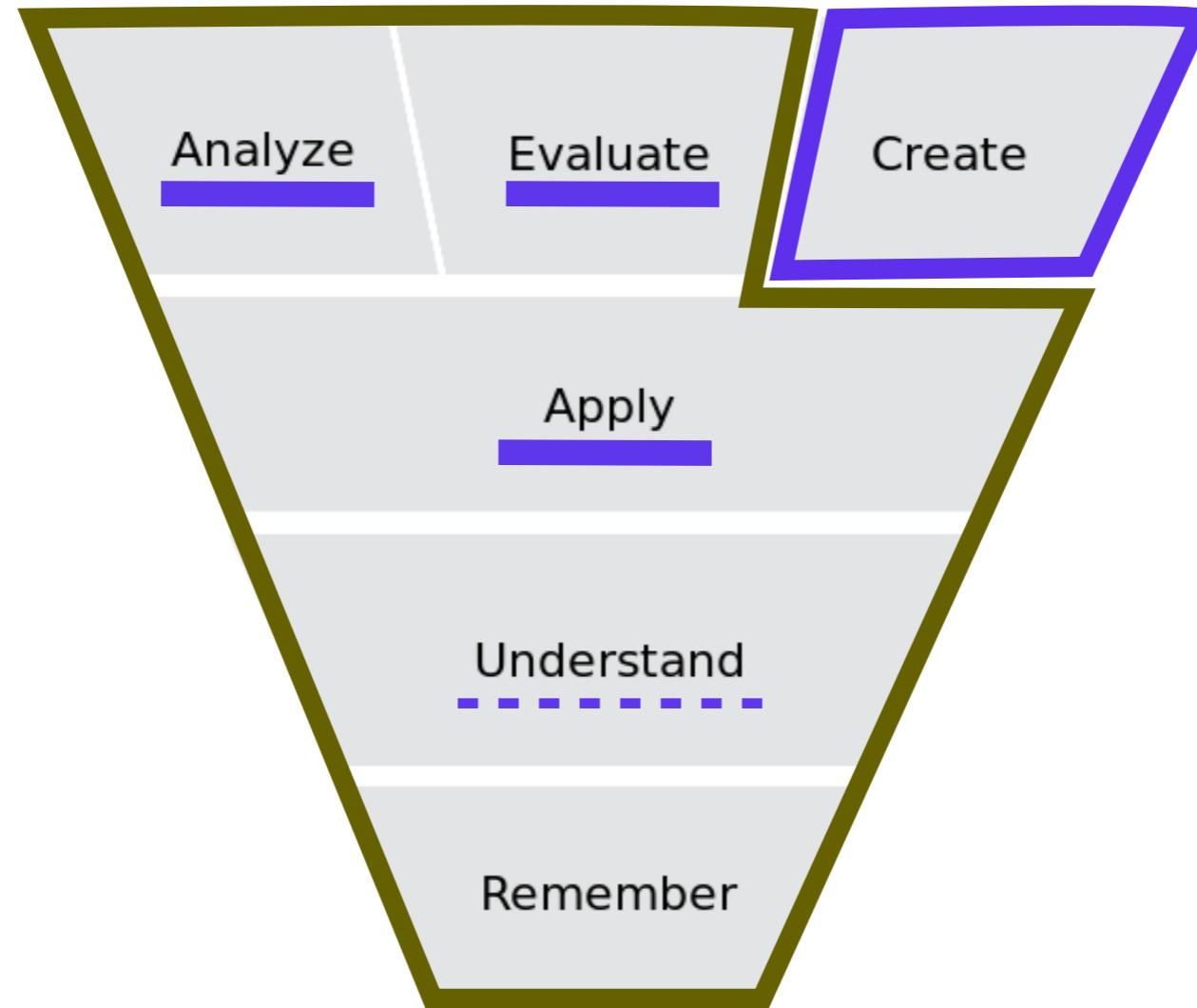
The project is open ended and each group must write a project proposal that must be approved by the teaching staff.

Writing a good project proposal is part of the learning process.

Developing skills

In an open ended group project you will practice and develop several personal skills including:

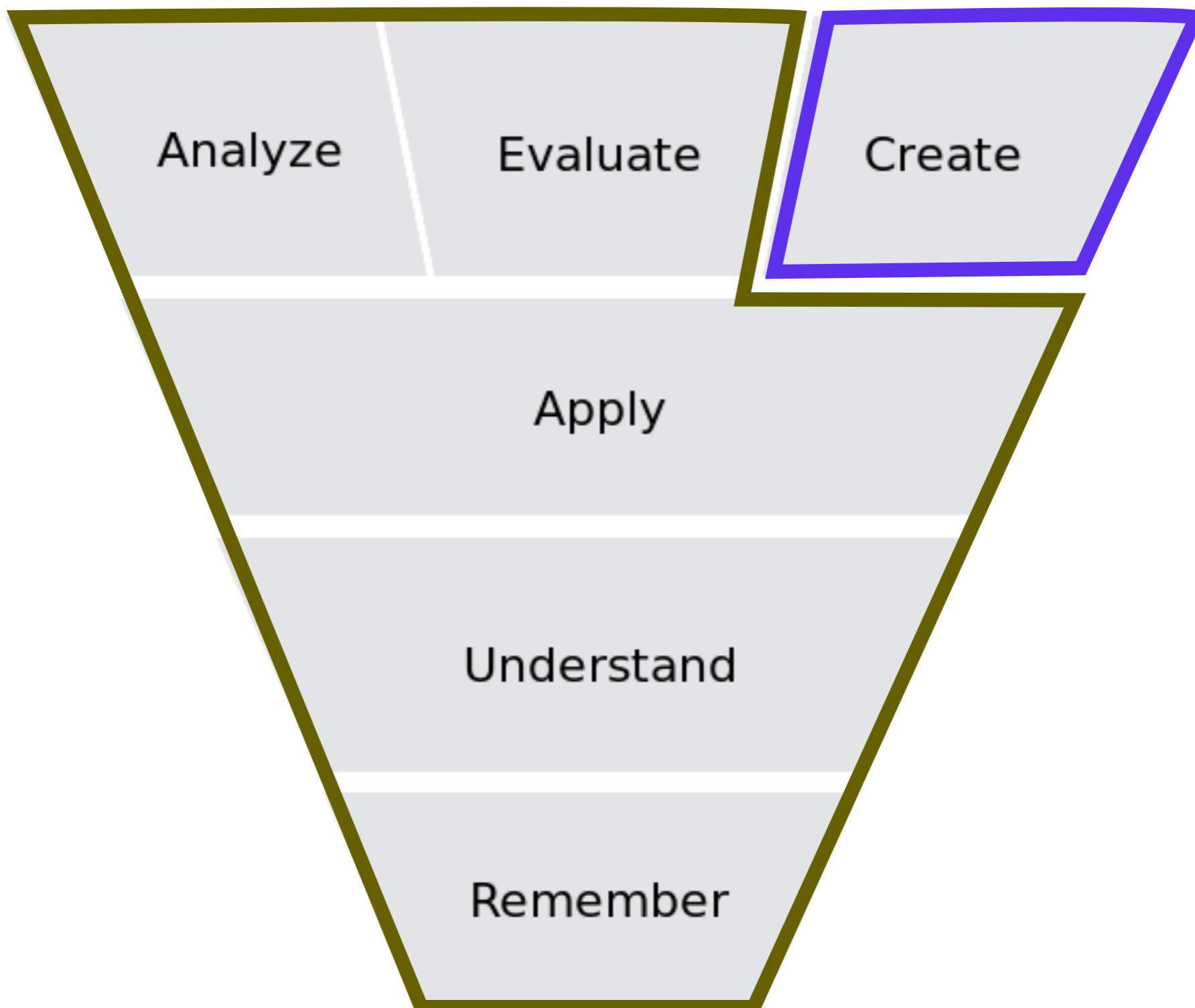
- **Collaboration:** within your project team.
- **Critical thinking:** asking the right questions to find a solution.
- **Creative thinking:** finding unique solutions to critical questions.
- **Planning time management:** finishing solutions when they are due.
- **Communication:** within your project team, presentation skills, writing skills.
- **Self-directed learning:** find and learn what you need.



By defining your own project, you are forced to reflect on the course content and analyze and evaluate your project in context of the course content.

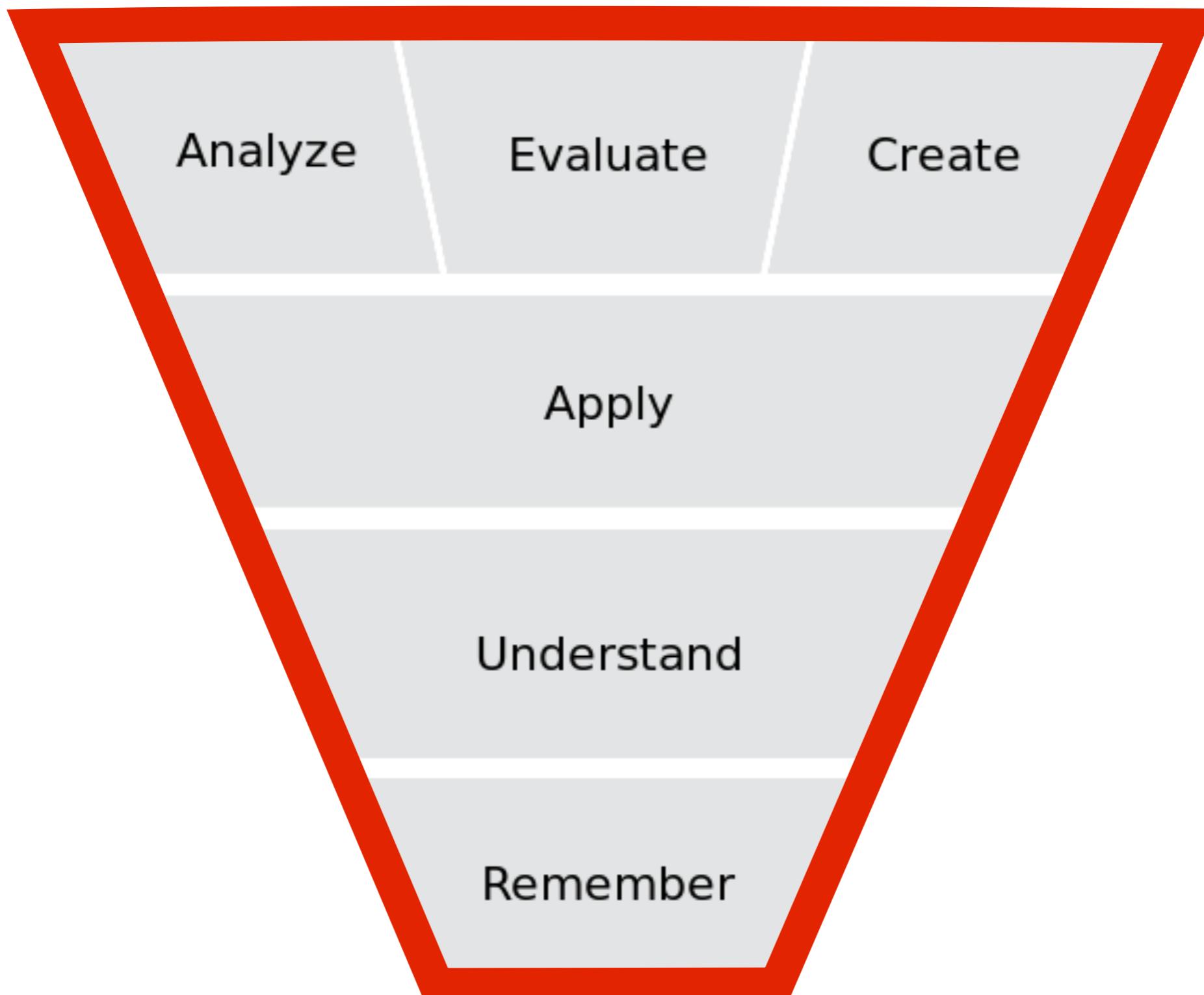
Categories in the cognitive domain of Bloom's Taxonomy

A consequence of the open ended group project is that you are forced to explore many of the categories of the cognitive domain.

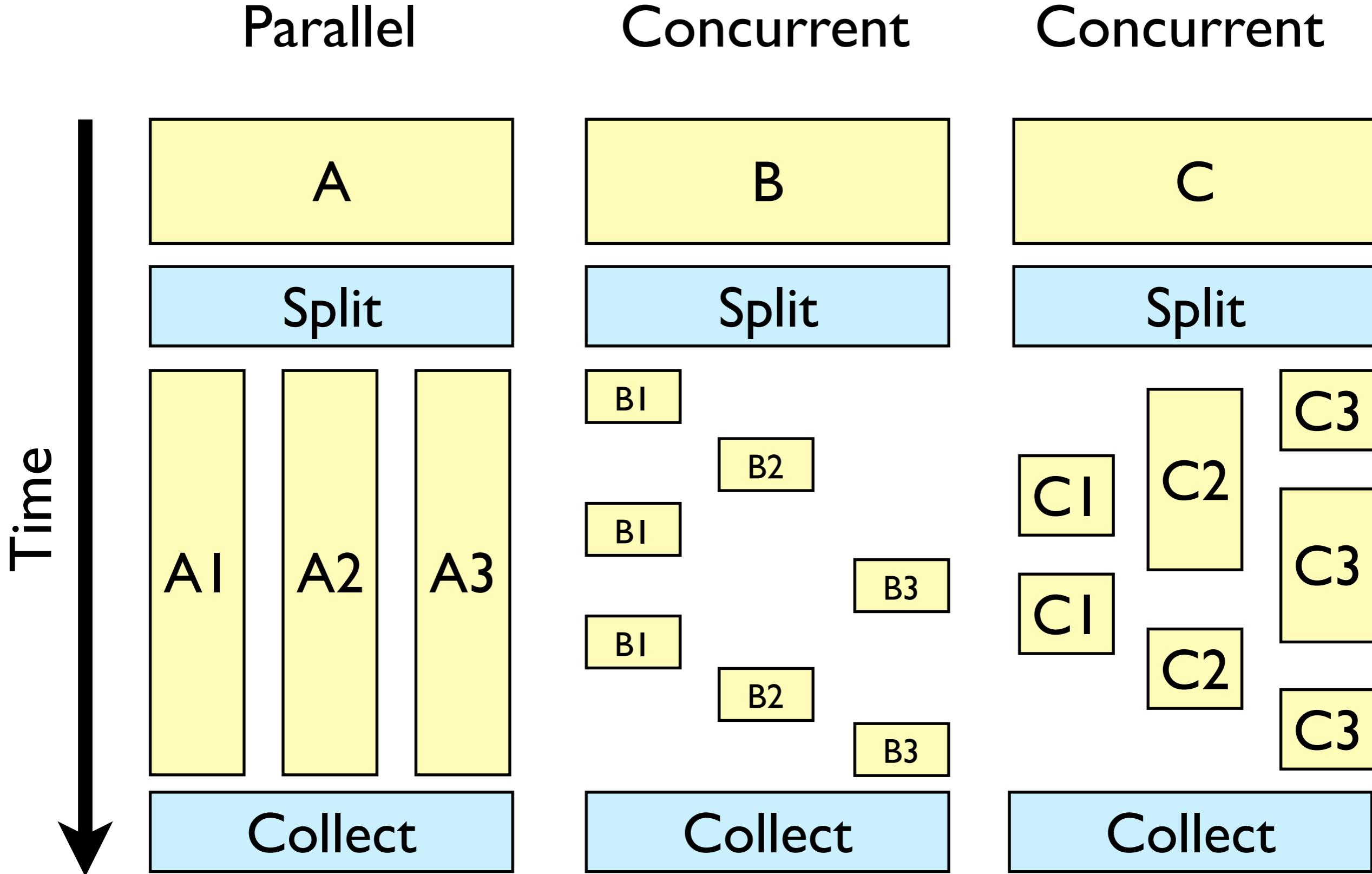


Categories in the cognitive domain of Bloom's Taxonomy

A consequence of the open ended group project is that you are forced to explore many of the categories of the cognitive domain.



Concurrency



Project Requirement

Project requirement

There is one important constraint - the **main theme** of the project must be **concurrency** and **concurrent programming**.

As long as the main focus of your project is related to concurrency and concurrent programming you are left with great **freedom** to design your project according to your interests and other preferences.

Concurrency models and programming languages

In the course we have studied several concurrency models and programming languages.

- Preemptive multitasking with non-shared memory using **POSIX processes** in C.
- Preemptive multitasking with shared memory using **Pthreads** in C.
- Asynchronous message passing with **actors** in Erlang.

You are not restricted to any of the above models and programming languages. You are free to use any programming language you see fit and are encouraged to use more than one programming language.

**Studietakt
och
arbetsinsats**

Studietakt

Vi är nu inne i vecka 13 och vårterminen och projektet slutar vecka 23.

Studietakt				
Period	Veckor	Hp	%	Timmar/vecka
3	4 - 12	5	33	13
4	13 - 23	10	67	27

Bedömnings

Hur bedöms och betygssätts
projektet?

Gruppen och projektet bedöms

- **Främst** utifrån hur väl projektet relaterar till kursens lärandemål med särskilt vikt vid concurrency (samtidighet).
- **Inte** utifrån hur väl slutresultatet uppfyller det ursprungliga projektförslaget.
- **Löpande** efter hur uppsatta delmål uppnås.

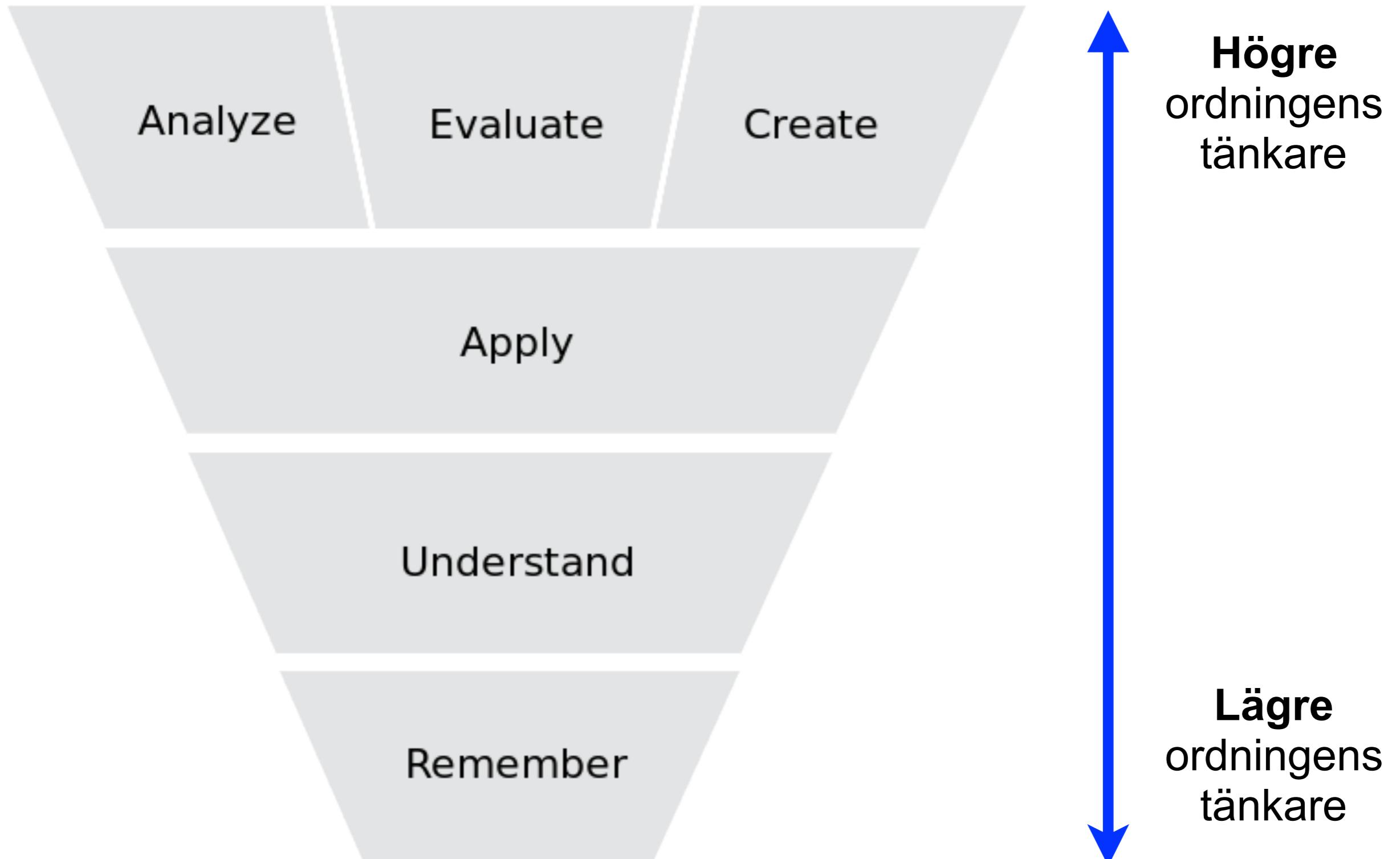
Individer bedöms

- **Främst** utifrån hur väl individen uppvisar förståelse för kursens lärandemål och hur de relaterar till projektet, särskild vikt läggs vid concurrency (samtidighet).
- **Utifrån** hur väl individen i detalj kan redogöra för sitt eget bidrag till projektet.
- **Även** utifrån sin övergripande förståelse och förmåga att redogöra för projektet som helhet.

Graderad

bedömnings

För högre betyg måste du **återkommande** under projektet visa förmåga till **högre ordningens tänkande**. För högre betyg behöver du återkommande visa att du befinner dig i de övre delarna av Blooms kognitiva domän.



Exempel på

graderad

bedömning

Notera att följande exemplet rör sig om just ett exempel och att ni inte skall försöka uppfylla just detta exempel eller använda exemplet som en exakt mall. Använd istället exemplet som inspiration och försök hitta så många tillfällen som möjligt att uppvisa högre ordningens tänkande.

Exempel på områden där du kan försöka uppvisa högre ordningens tänkande:

- Val av programmeringsspråk (fokus på concurrency).
- Designval (algoritmer, datastrukturer, systemarkitektur, mm).
- Utvärdering av resultat.
- Analys av kluriga buggar relaterade till concurrency.

Listan ovan är inte sorterad i prioritetsordning. Listan är inte heller heltäckande.

Exempel på Graderad bedömnings

Ni har bestämt er förr att använda en viss
lösningsmetod/algoritm.

Betyg 3: Ni berättar om metoden ni valt och motiverar ert val:

- *Den var lätt att implementera.*
- *Den löser vårt problem.*
- *Vi kände till metoden sedan tidigare.*

Betyg 3/4: Ni motiverar ert designbeslut genom att redogöra för metodens lämplighet i er implementation:

- *I vår implementation behövs en lösning som tillhandahåller X vilket metod A gör.*

Betyg 4/5: Ni jämför med andra metoder och deras lämplighet:

- *Metod B tillhandahåller också X men är inte lika effektiv under förutsättning F.*

Betyg 5: Ni redogör även för konsekvenser av ert val och jämför med konsekvenser av andra val

- *Genom att använda metod M1 hamnar vi i situation S1 där problem P1 måste lösas. Om vi använt metod M2 skulle vi i stället hamnat i situation S2 med problem P2. Problem P2 är lättare att lösa än problem P1, men vi anser att den effektivitet vi uppnår genom att använda metod M1 är viktigare.*

Ni har bestämt er förr att använda en viss lösningsmetod/algoritm.

Betyg 3: Ni berättar om metoden ni valt och motiverar ert val (t ex. "den var lätt att implementera", "den löser vårt problem", "vi kände till metoden sedan tidigare").

Betyg 3/4: Ni motiverar ert designbeslut genom att redogöra för metodens lämplighet i er implementation (t ex. "i vår implementation behövs en lösning som tillhandahåller X vilket metod A gör").

Betyg 4/5: Ni jämför med andra metoder och deras lämplighet (t ex. "metod B tillhandahåller också X men är inte lika effektiv under förutsättning F")

Betyg 5: Ni redogör även för konsekvenser av ert val och jämför med konsekvenser av andra val (t ex. Genom att använda metod A hamnar vi i situation S1 där problem P1 måste lösas. Om vi använt metod B skulle vi i stället hamnat i situation S2 med problem P2. Problem P2 är lättare att lösa än problem P1, men vi anser att den effektivitet vi uppnår genom att använda metod A är viktigare.)

How to get higher grades?

Examples of opportunities to show that you are worth a higher grade:

- Design choices (good motivations)
- Evaluation of results
- Analysis of problems you encounter during the project
- Clever and/or creative solutions
- ...

What is assessed? When?

- Everything that a teacher sees is assessed.
 - ▶ However – the course is a learning opportunity, not a test.
- In practice:
 - ▶ Teacher takes notes of everything positive (and nothing negative) throughout the course.
 - ▶ Grading is done after the final submission – all teachers participate.
- **Please** – ask questions about grading whenever you feel unsure!

Exam code			Hp	Grading scale	Comment
Nr	Name	Symbol			
1000	Theory (part 1)	T ₁	2	Fail/Pass	
2000	Assignments (part 1)	A ₁	2	Fail, 3, 4, 5	This grade is calculated by rounding the average grade of the three programming assignments for module 1, 2 and 4.
3000	Written exam (part 1)	E	1	Fail, 3, 4, 5	
4000	Theory (part 2)	T ₂	1	Fail/Pass	
5000	Assignments (part 2)	A ₂	1	Fail, 3, 4, 5	This grade is determined by the grade programming assignment for module 8.
6000	Project (group)	P _g	2	Fail, 3, 4, 5	
7000	Project (individual)	P _i	5	Fail, 3, 4, 5	
0400	Project report	P _r	1	Fail/Pass	

Slutbetyg på kursen

The final grade (FG) is computed by rounding the weighted average of the assignments and the project.

$$FG = \left[\frac{2A_1 + A_2 + 5P_i + 2P_g}{10} \right]$$

Varje par generar några olika förslag

Varje par har i uppgift att översiktligt formulera minst två egna projektförslag innan projektgrupperna bildas.

Varje individ

Varje individ skall ta fram en egen lista med sådant den är intresserad av att lära sig mer om under projektet.

Exempel

- Modeller för concurrency
- Specifika programmeringsspråk
- Specifika problem och/eller algoritmer
- Design och systemarkitektur
- Test och debugging
- Olika roller i ett projekt
- Skriftlig framställning
- Muntlig framställning
- Annat du är intresserad av

Gruppen väljer ett förslag

När projektgruppen bildas måste gruppen komma överens om ett projektförslag. Det kan vara något förslag som något av paren arbetat fram sedan tidigare eller ett helt nytt förslag.



*Vi vet ju inte hur stort
projektet är och vilka
avgränsningar som skall
göras för att hinna klart i
tid vilket gör det svårt att
skriva ett projektförslag.*

Syfte

Syftet med arbetet att ta fram olika
förslag och sedan välja ett är främst att ni
skall få jobba med något ni verkligen är
intresserade av.

Exempel

Exempel på hur översiktliga
projektförslag (idéer) kan
formuleras.

Två exempel på tänkbara projektförslag och hur de kan formuleras.

- (a) Ett system för reservation och försäljning av biobiljetter.
- (b) Simulering där autonoma enheter rör sig i en virtuell värld och tillsammans bygger upp en karta över denna värld.

Notera att båda dessa projektförslag är formulerade så att det går att variera i omfattning och komplexitet.

Avgränsningar

När gruppen hittat något de vill jobba med utgår vi (Karl och Aletta) från ert projektförslag och gör lämpliga avgränsningar tillsammans med er.

Projektförslaget är alltså inte tänkt som den slutgiltiga projektplanen utan som ett första steg.

Avgränsningar och delmål

**Genom regelbundna möten gör vi
tillsammans lämpliga avgränsningar och
sätter delmål till nästa möte.**

• **group project** /noun/:

time to relax while you watch someone who cares, do all the work.

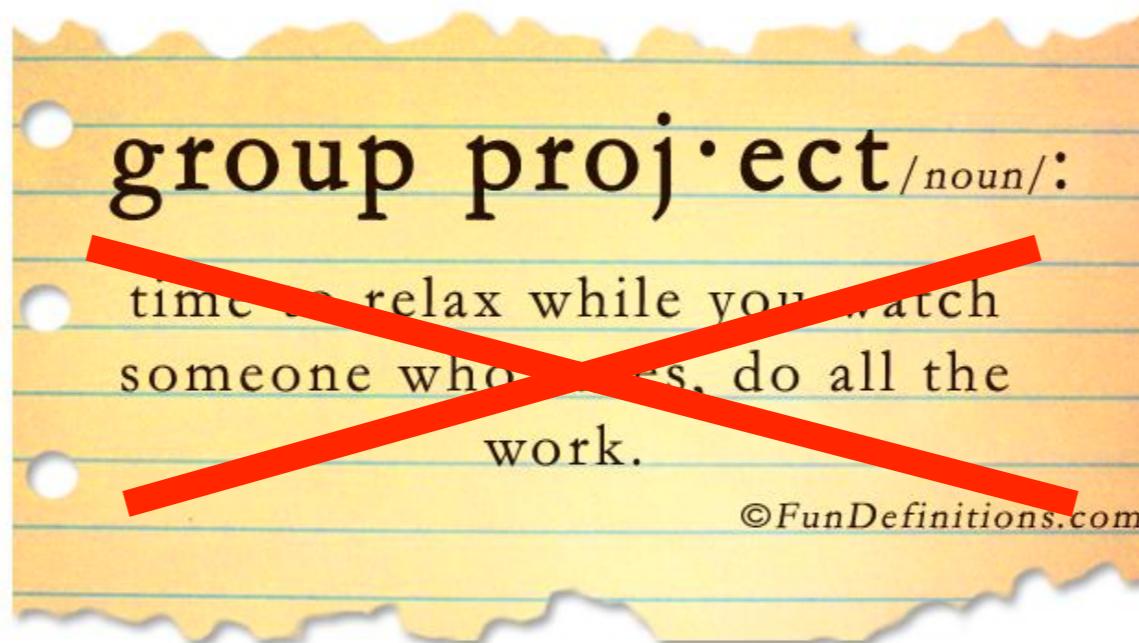
©FunDefinitions.com

Mål med kursen (2)

Från kursplanen för kursen Operativsystem och processorienterad programmering (1DT096).

Efter godkänd kurs ska studenten kunna:

- **använda** verktyg för **versionshantering**, **felsökning** och **testning**.
- **genomföra** ett **strukturerat projektarbete** tillsammans med andra studenter samt, under processen, **reflektera** över och utveckla gruppens samarbete.
- **planera** ett **projekt**, inklusive användande av samarbetsverktyg, så att det kan genomföras inom givna ramar.
- **presentera och diskutera** **kursens innehåll muntligt** och **skriftligt** med för utbildningsnivån lämplig färdighet.



- Projektet skall vara ett samarbete mellan gruppens medlemmar.
- Alla skall bidra till slutresultatet.
- Alla behöver inte göra samma saker.
- Var och en skall i detalj kunna redovisa för sitt bidrag till projektet.
- Var och en skall kunna redogöra för hur projektets olika delar samverkar.

Redovisning av gemensamt projektförslag

Efter att gruppen kommit överens om ett gemensamt projektförslag skall det **redovisas** både **muntligt** och **skriftligt**.

Systemarkitektur

En viktig del av det gemensamma
projektförslaget är ett första försök till
en systemarkitektur.

To be able to reason about the proposed system you must draft a system architecture.

- ▶ The purpose of the system architecture is to communicate the overall structure of the system.
- ▶ A good system architecture is needed in order to break down the system in smaller pieces.

System architecture example (1)

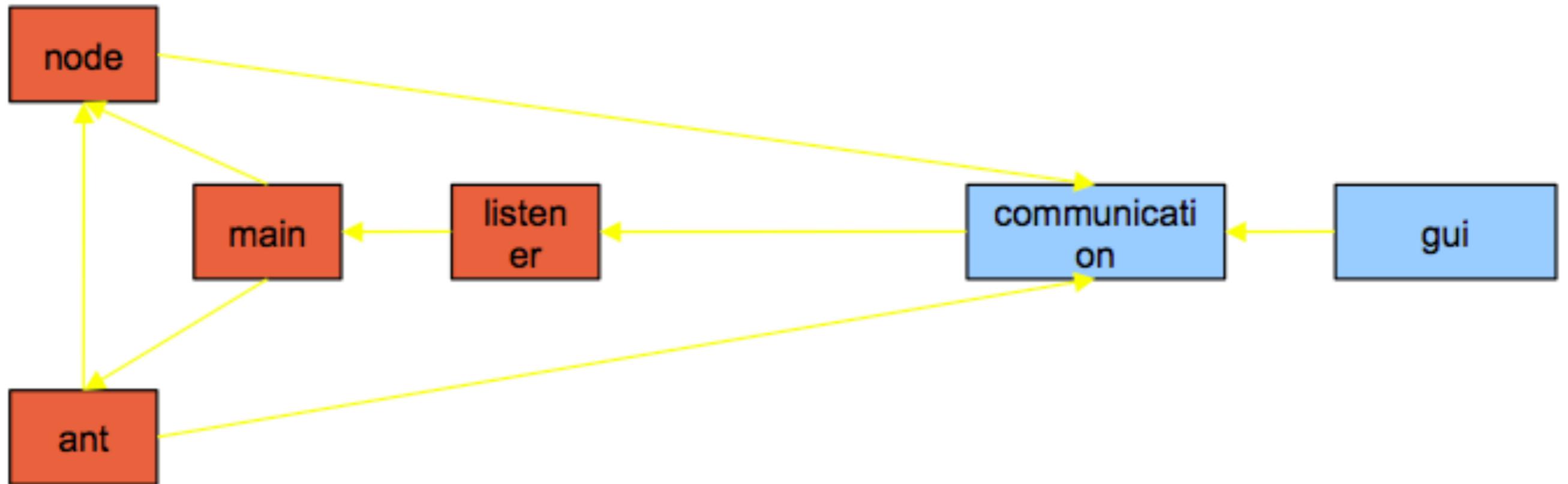
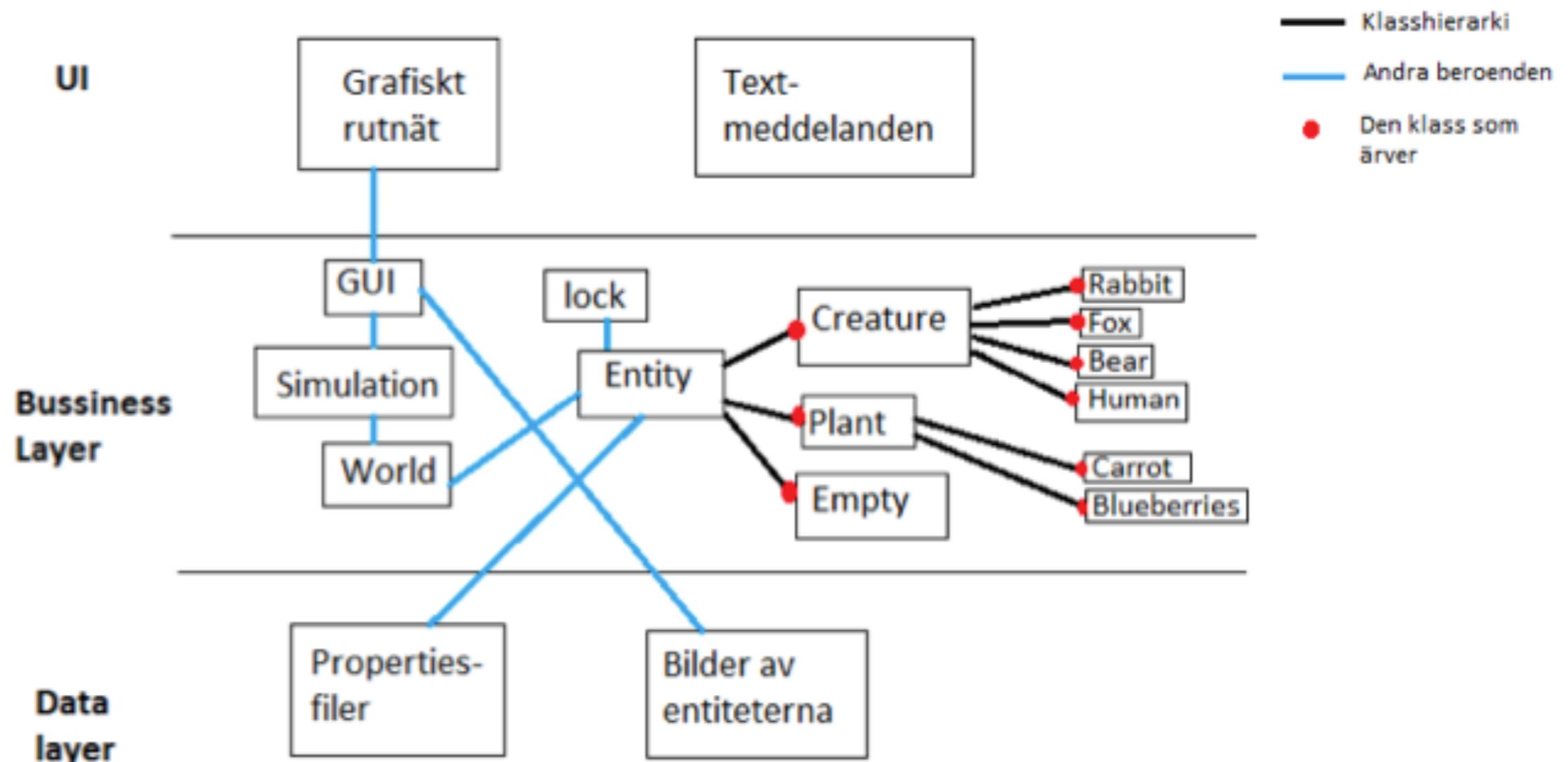


Fig 3. Systemarkitektur, översikt (orange: back-end (Erlang), blå: front-end (Python)).

A hybrid system using both Erlang and Python is described

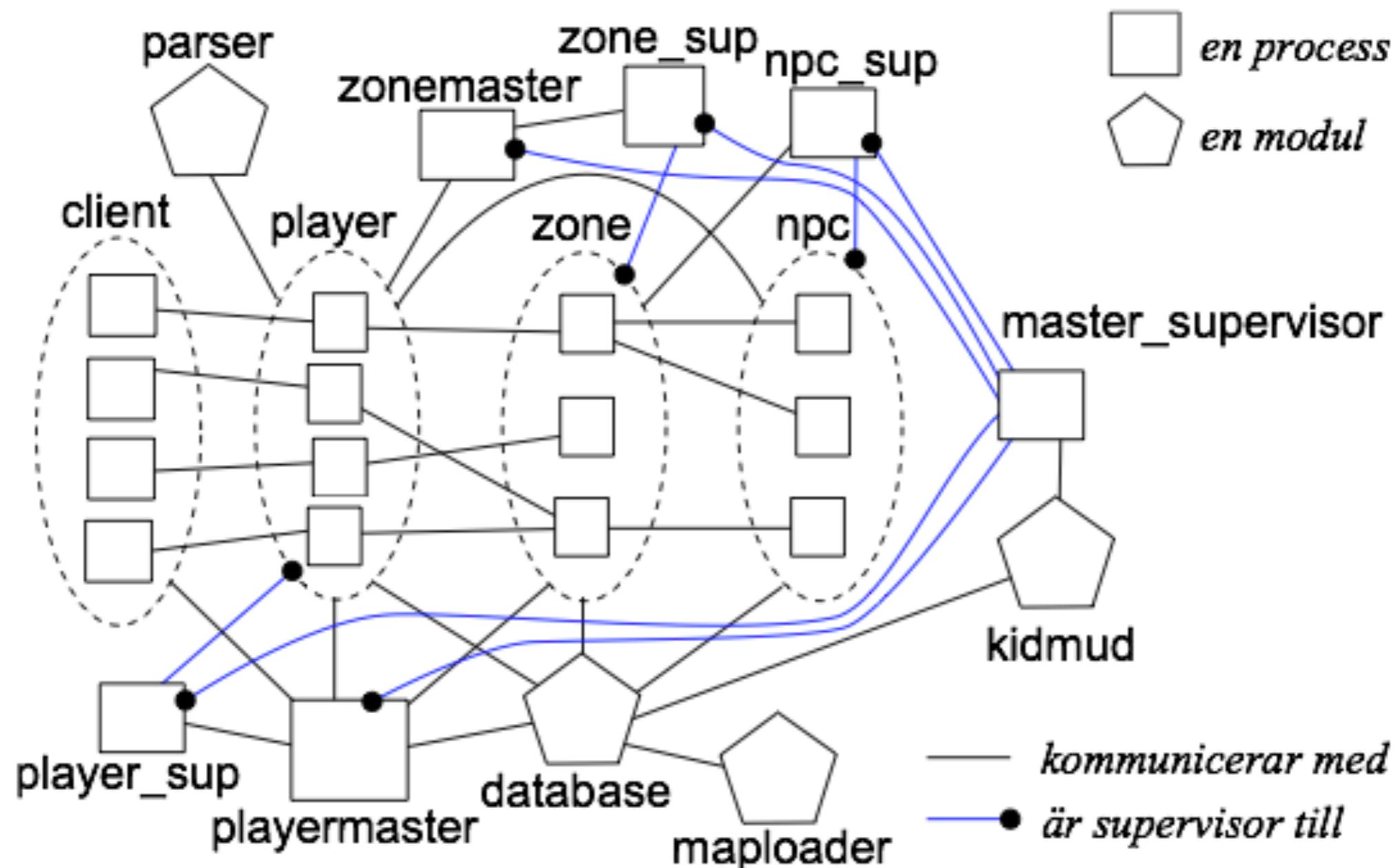
System architecture example (2)



Figur 2. Schematisk bild av Java-systemet

A three-tier architecture is used to separate the system into three categories: user interface, logic and data.

System architecture example (3)



Figur 2: Moduler som kommuniceras och supervisorträd

A system using Erlang is described focusing on dependencies between processes and modules.



Hemsidan

Mer information om projektet kommer
löpande publiceras på kursens hemsida

-  9 - Group project
-  Kom igång
-  Timeline
-  Bedömning

Kom igång



9 - Group project



Kom igång



Timeline



Bedömning

Kom igång

Kursen Operativsystem och processorienterad programmering (1DT096) avslutats under period 4 med ett projekt.

Uppstart

Projektdelen av kursen inleds med en föreläsning torsdagen 28/3 med övergripande information om projektet.

Projektenkät

Inför projektet måste du svara på en [enkät](#).



OSPP (1DT096) 2020 - Projektenkät

Kursen avslutas med ett större projektarbete i grupp (8 Hp). Under projektet kommer ni arbeta i grupper om 6 studenter. De par om två studenter ni arbetat i tidigare kommer i sw långt det är möjligt att hållas ihop under projektet och baserat på svaren i denna enkät sättas samman till projektgrupper med andra par.

*Required

Deadline: fredag 27/3 kl | 3:00

Detaljerad projektplan



9 - Group project



Kom igång



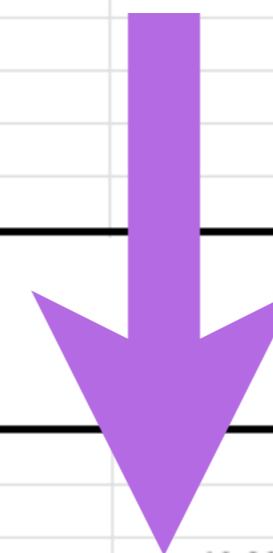
Timeline



Bedömning

Period 4

W	D	Datum	Bivillkor	Aktivitet	Obligatoriskt (examination)	Tid	Lärare	Rum (Zoom eller fysikt)	Beskrivning
13	Mo	2020-03-23							
	Tu	2020-03-24							
	We	2020-03-25							
	Th	2020-03-26		Lecture			Karl & Aletta	Zoom (kodpar)	Uppstart av projekt
	Fr	2020-03-27		Deadline	Ja (alla)	13:00			Projektenkät
14	Mo	2020-03-30		Workshop			Karl		Projektförslag
	Tu	2020-03-31		Workshop			Karl & Aletta		Process (gruppdynamik, roller)
	We	2020-04-01		Presentation	Ja (alla)	Helklass, gärna 2 timmar före och 2 timmar efter lunch	Karl & Aletta		Muntlig presentation av projektförslag
	Th	2020-04-02		Deadline	Ja (grupp)	12:00	Aletta		Inlämning av skriftligt projektförslag (draft)
	Fr	2020-04-03		Workshop	Ja (minst 2 ur gruppen)		Aletta	Zoom	Skriftligt projektförslag
15	Mo	2020-04-06	Påsklov, omtentor						
	Tu	2020-04-07							
	We	2020-04-08							
	Th	2020-04-09							
	Fr	2020-04-10							
22	Mo	2020-05-25							
	Tu	2020-05-26							
	We	2020-05-27		Tutoring			Karl		Handledning 4
	Th	2020-05-28		Deadline	Ja (grupp)	12:00			Skriftlig rapport - utkast 3
	Fr	2020-05-29		Presentation	Ja (alla)	Helklass, gärna 2 timmar före och 2 timmar efter lunch	Karl & Aletta		Slutpresentatuer av projekt
	Fr	2020-05-29		Deadline	Ja (grupp)	17:00			Kamratåterkoppling på skriftlig rapport
23	Mo	2020-06-01	Avlyst dag						
	Tu	2020-06-02	Tentamen						
	We	2020-06-03							
	Th	2020-06-04		Deadline	Ja (grupp och individuellt)	12:00			Skriftlig rapport, individuell reflektion, lärjournal
	Fr	2020-06-05		Seminar	Ja (alla, gruppvis)		Karl & Aletta		Slutseminarium



W	D	Datum	Bivillkor	Aktivitet	Obligatoriskt (examination)	Tid	Lärare	Rum (Zoom eller fysikt)	Beskrivning
13	Mo	2020-03-23							
	Tu	2020-03-24							
	We	2020-03-25							
	Th	2020-03-26		Lecture			Karl & Aletta	Zoom (kodad)	Uppstart av projekt

Vi kommer försöka hålla oss till denna planering så långt det är möjligt. Förändringar kan komma att göras löpande vid behov.

W	D	Datum	Bivillkor	Aktivitet	Obligatoriskt (examination)	Tid	Lärare	Rum (Zoom eller fysikt)	Beskrivning
Th		2020-06-04							
Fr		2020-06-05		Seminar	Ja (alla, gruppvis)		Karl & Aletta		Slutseminarium

Vad händer
nu?

Varje par generar några olika förslag

Varje par har i uppgift att översiktligt formulera minst två egna projektförslag innan projektgrupperna bildas.

Varje individ

Varje individ skall ta fram en egen lista med sådant den är intresserad av att lära sig mer om under projektet.

Exempel

- Modeller för concurrency
- Specifika programmeringsspråk
- Specifika problem och/eller algoritmer
- Design och systemarkitektur
- Test och debugging
- Olika roller i ett projekt
- Skriftlig framställning
- Muntlig framställning
- Annat du är intresserad av

Nya projektgrupper

På workshopen måndag 30/3
kommer ni få reda på
projektgruppernas sammansättning.

Gruppen väljer ett förslag

När projektgruppen bildats måste gruppen komma överens om ett projektförslag. Det kan vara något förslag som något av paren arbetat fram sedan tidigare eller ett helt nytt förslag.