

Project kickoff

Module 9

Lecture

Operating systems and process oriented programming 2019

1DT096

**Mål med
kursen**

Mål med kursen

(1)

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

Efter godkänd kurs ska studenten kunna:

- **redogöra** för hur **operativsystem** och **runtimesystem växelverkar** med **maskin-** och **programvara**.
- **förklara** 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 processororienterad 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.

Studenten och arbetsgivaren

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

Studenten

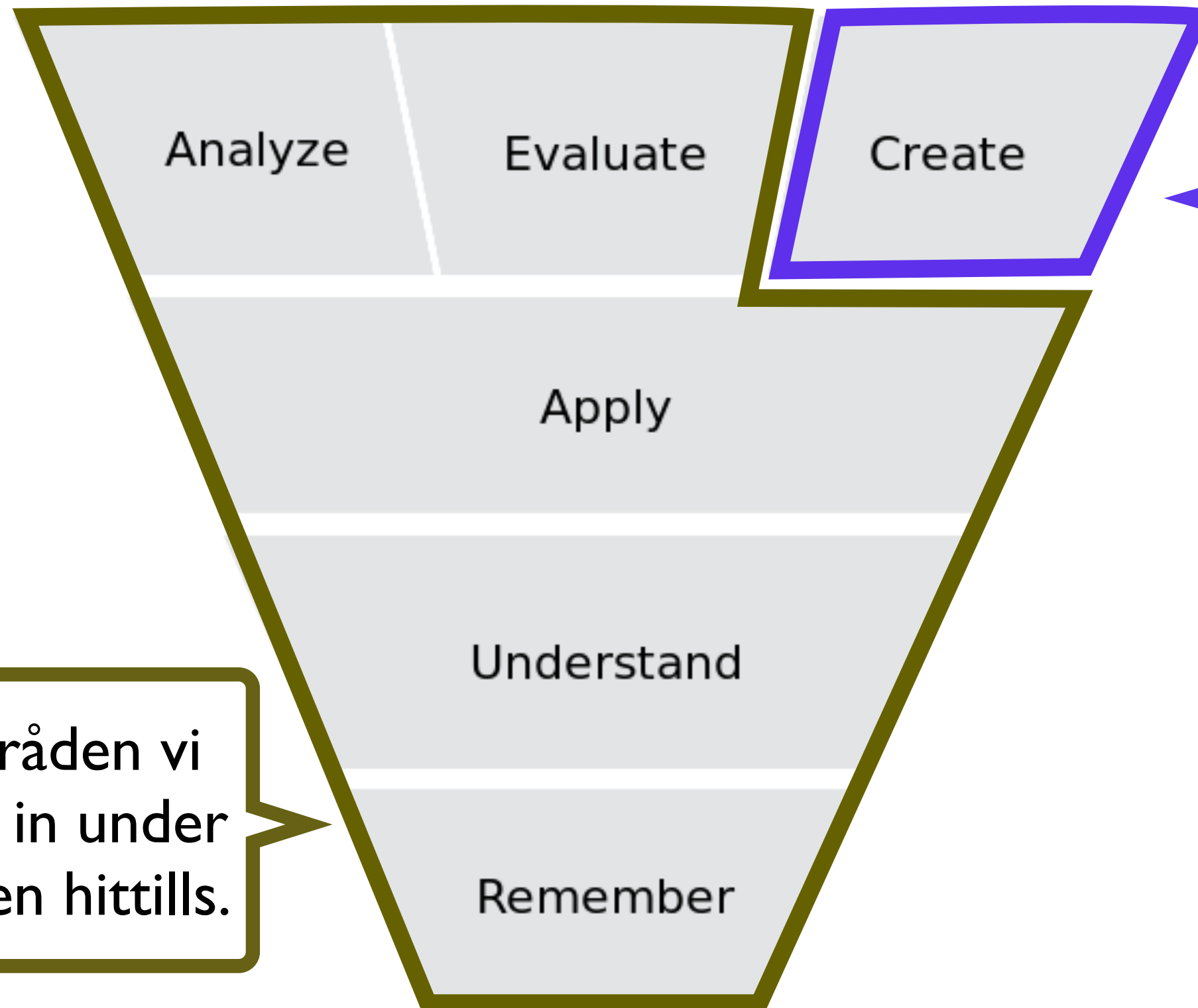
Jag kan väldigt mycket, men jag får inte tillräckligt väl definierade arbetsuppgifter.

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



Återstår att självständigt skapa något nytt.

Till viss del har denna del även täckts in av uppgifter för högre betyg.

Områden vi täckt in under kursen hittills.

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

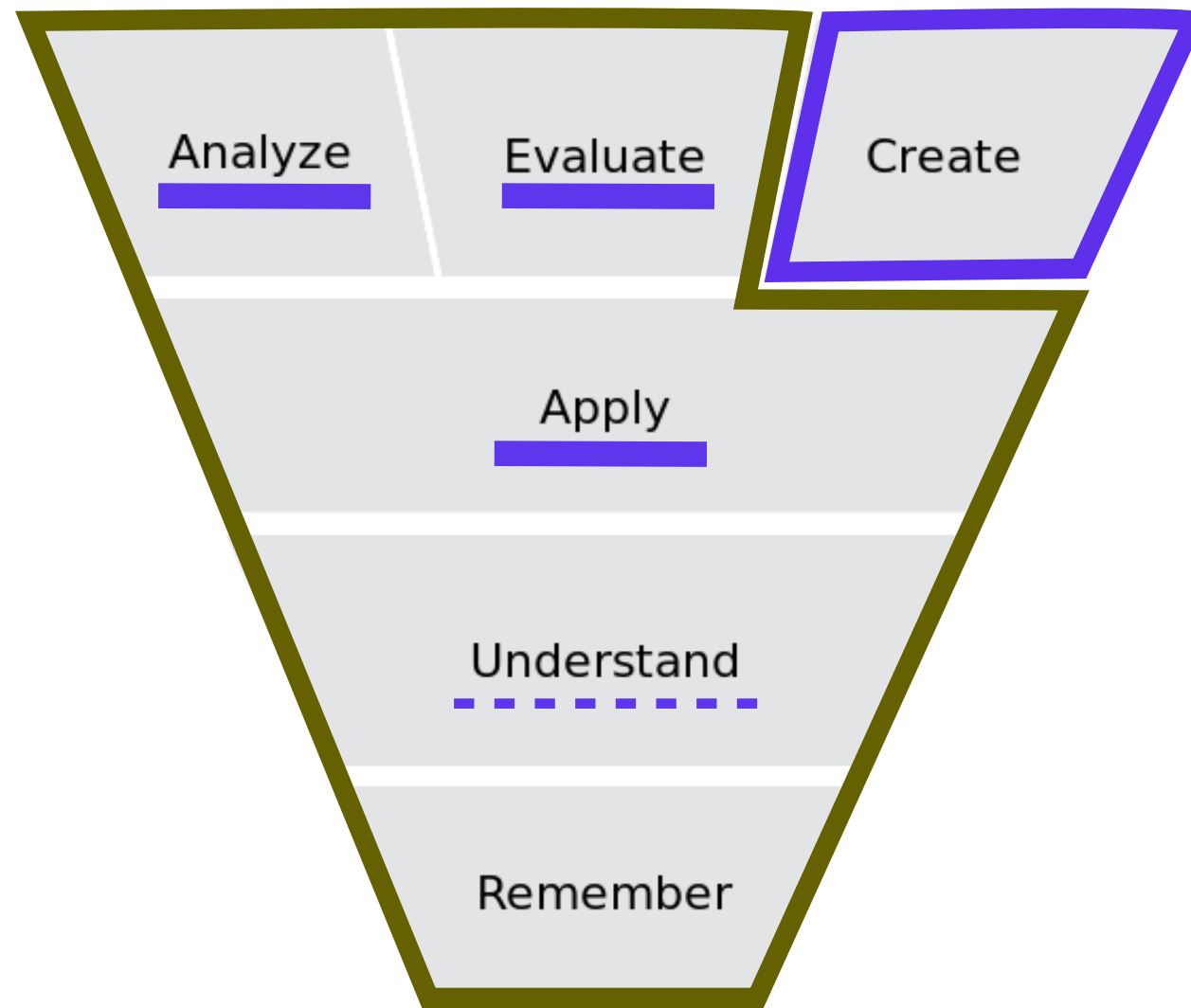
The project is open ended and each group must write a project proposal that must be approved by the teaching staff.

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:

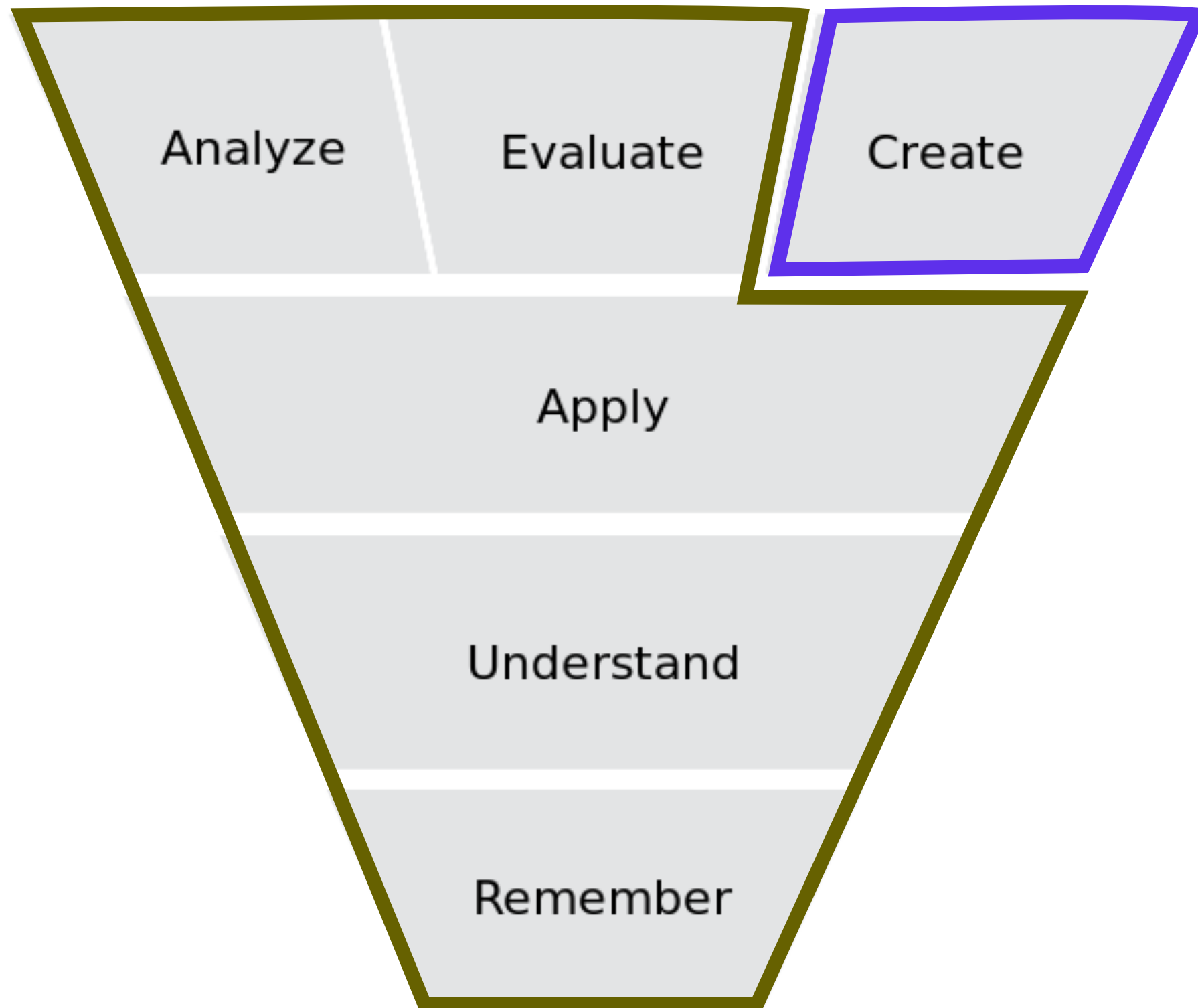
- **Critical thinking:** asking the right questions to find a solution.
- **Creative thinking:** finding unique solutions to critical questions.
- **Time management:** finishing solutions when they are due.



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

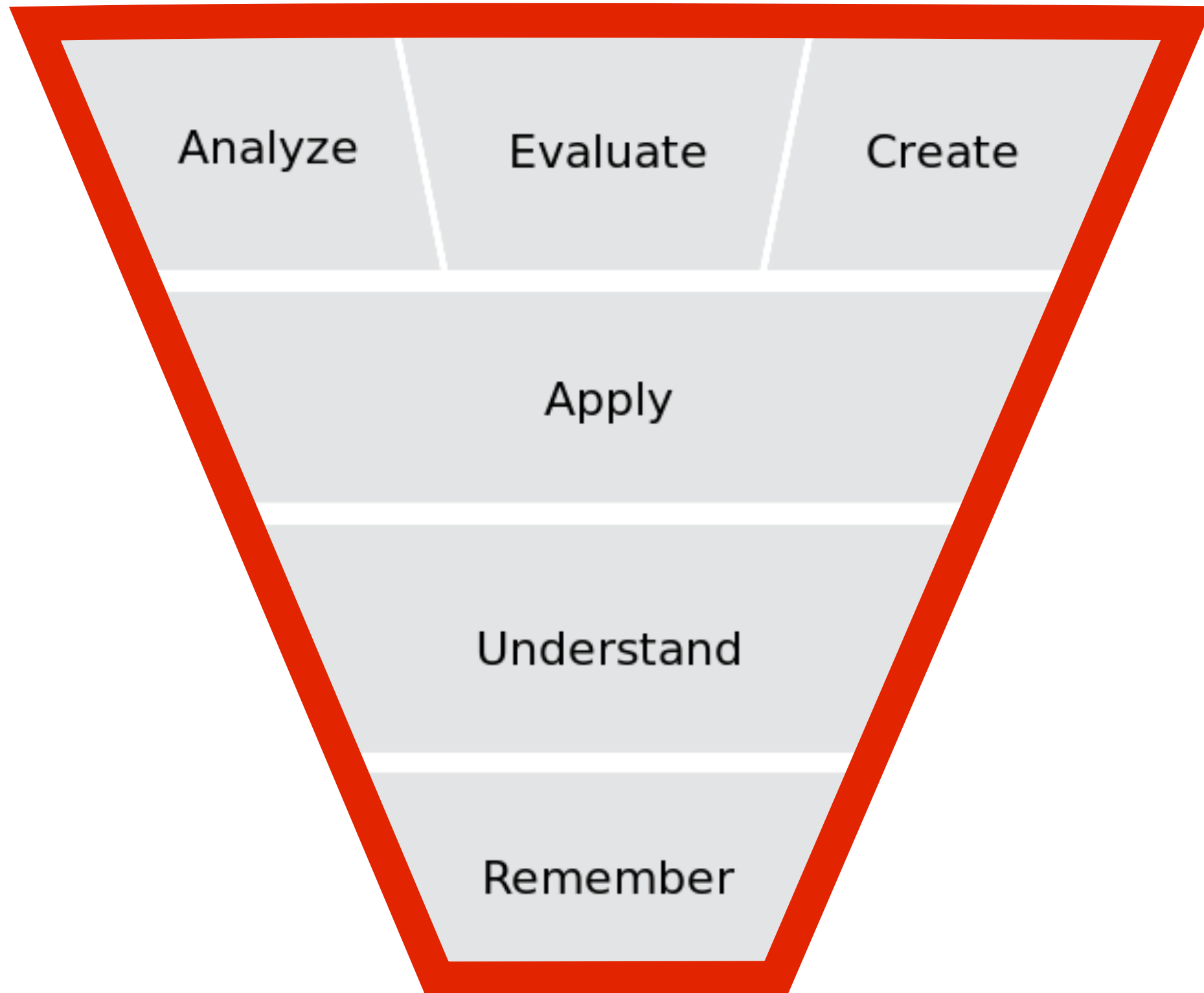
Categories in the cognitive domain of Bloom's Taxonomy

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

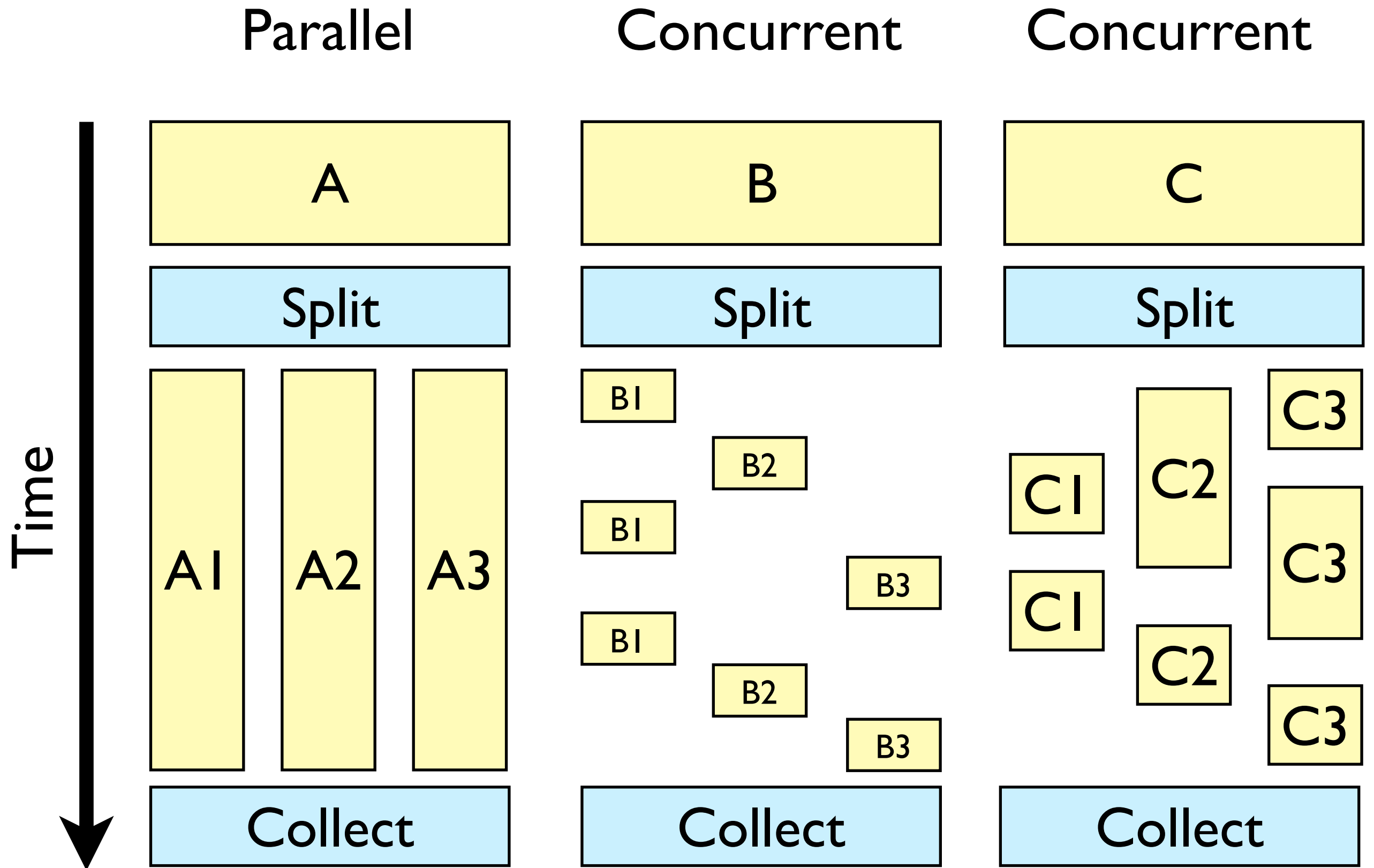


Categories in the cognitive domain of Bloom's Taxonomy

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

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

Bedömning

Hur bedöms och betygssätts projektet?

Projektet bedöms

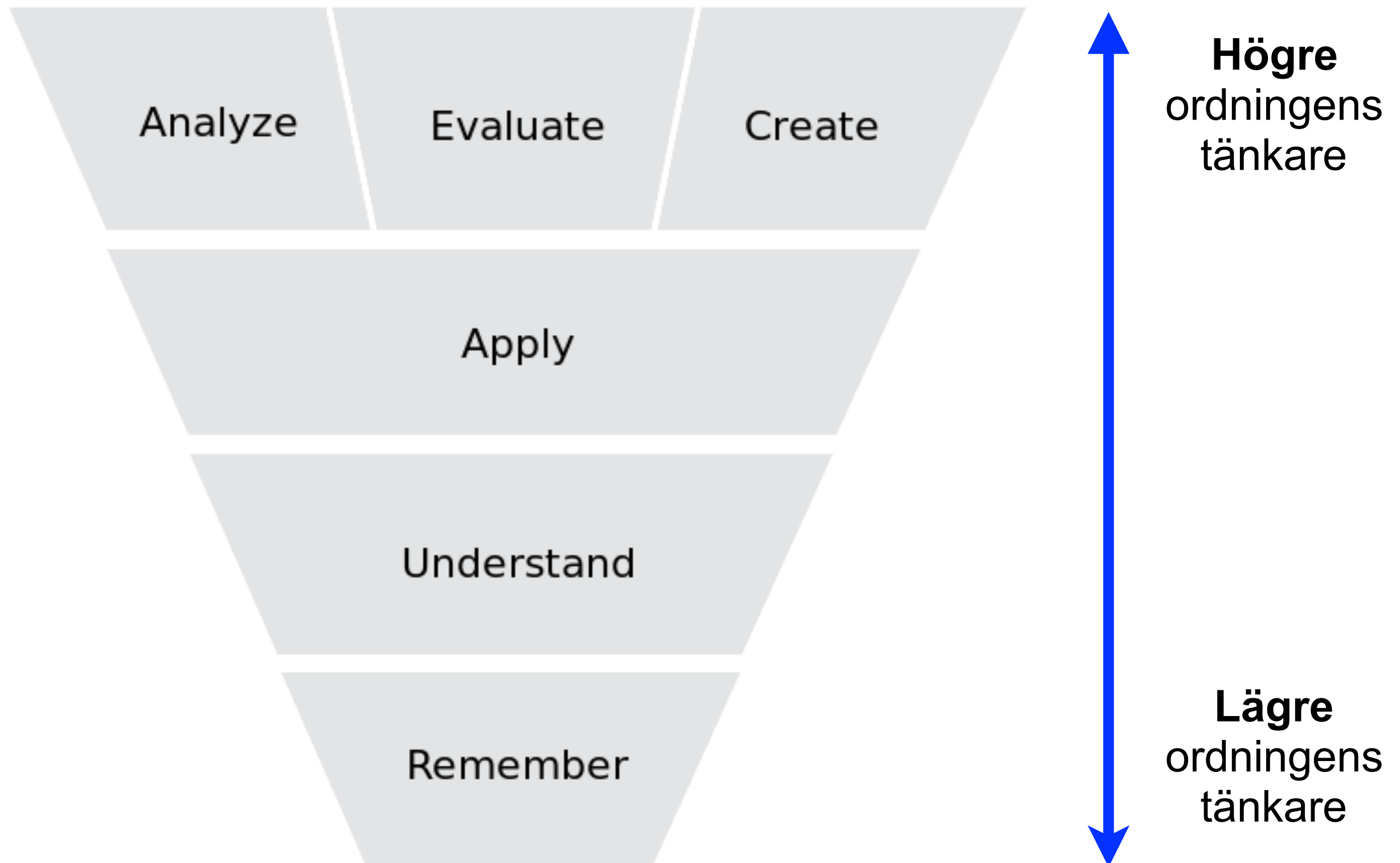
- **Främst** utifrån hur väl projektet relaterar till kursens lärandemål med särskilt vikt vid concurrency (samtidighet).
- **Inte** utifrån hur väl slutresultatet uppfyller det ursprungliga projektförslaget.
- **Löpande** efter hur uppsatta delmål uppnås.

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
bedömning**

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 bedömning

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

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/algorithm.

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.”)

Varje par generar några olika förslag

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

Varje individ

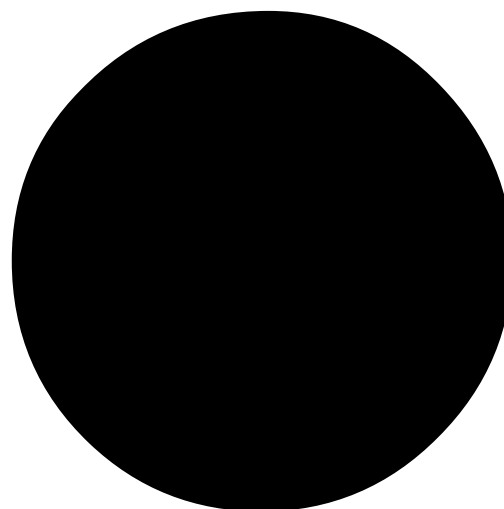
Varje individ skall ta fram en egen lista med sådant den är intresserad av att lära sig mer om under projektet.

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

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 ni gruppen hittat något de vill jobba med utgår vi 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 proj·ect */noun/*:

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

©*FunDefinitions.com*

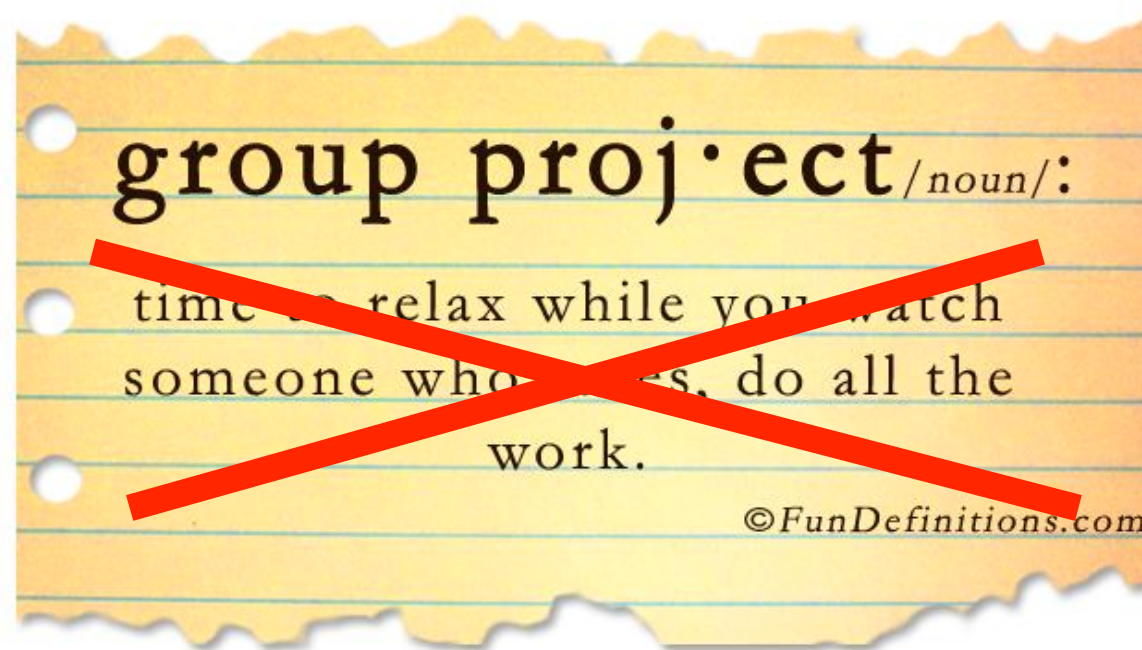
Mål med kursen

(2)

Från kursplanen för kursen Operativsystem och processororienterad 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.**

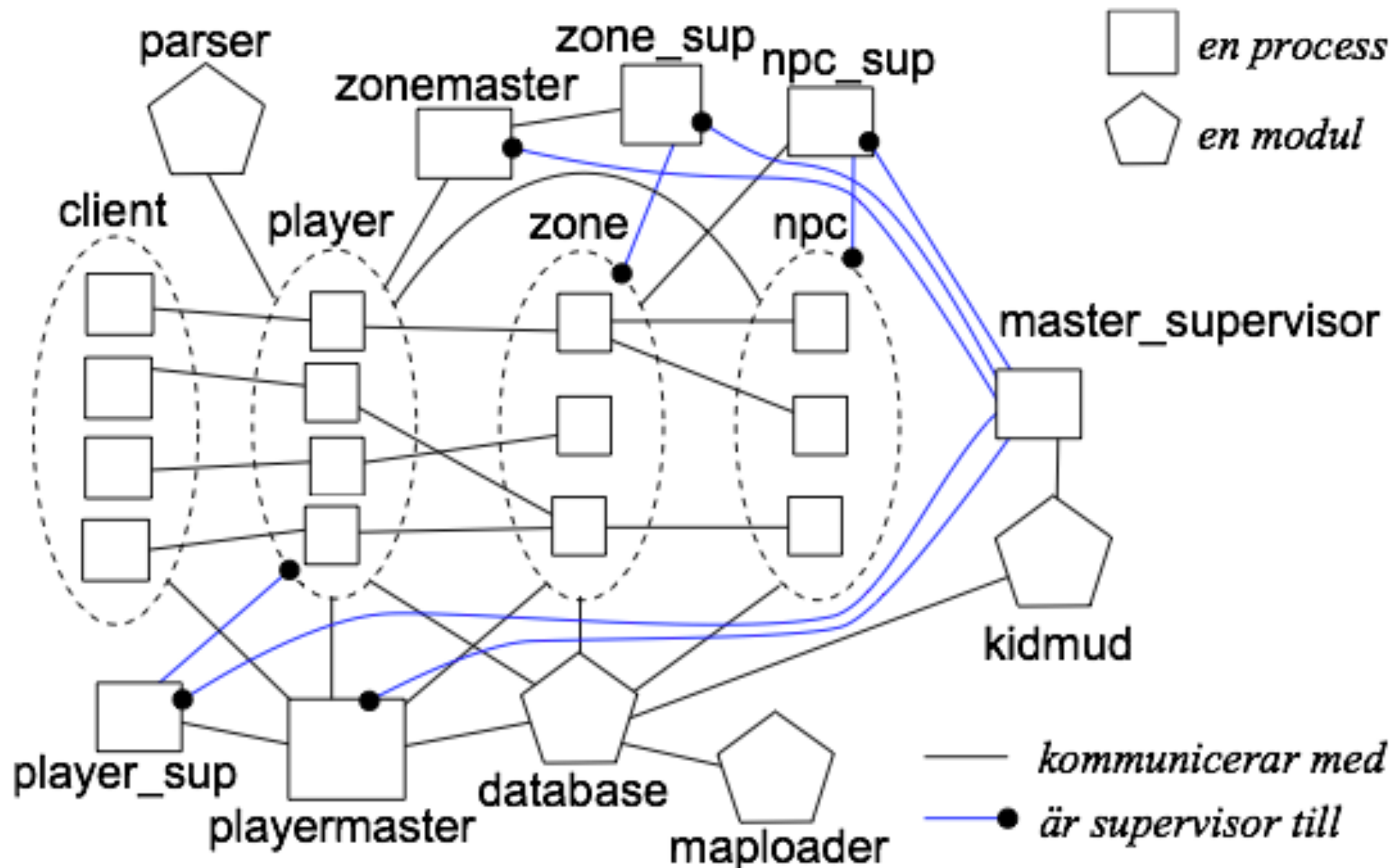
System- arkitektur

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)



Figur 2: Moduler som kommunicerar och supervisorträd

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

System architecture example (2)

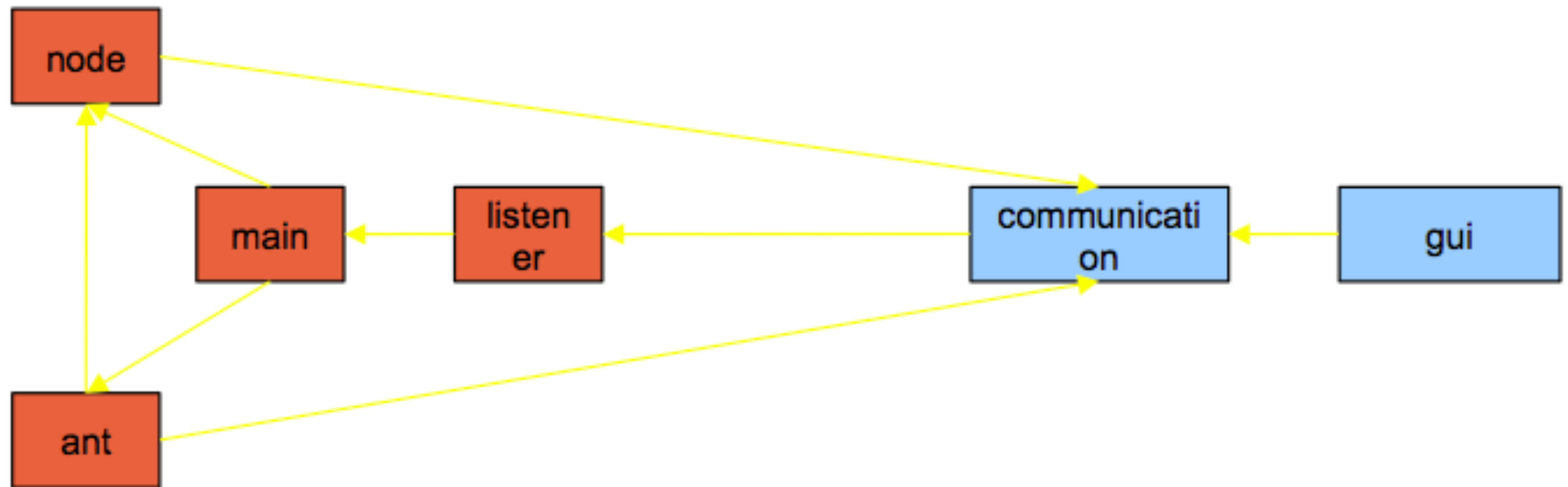
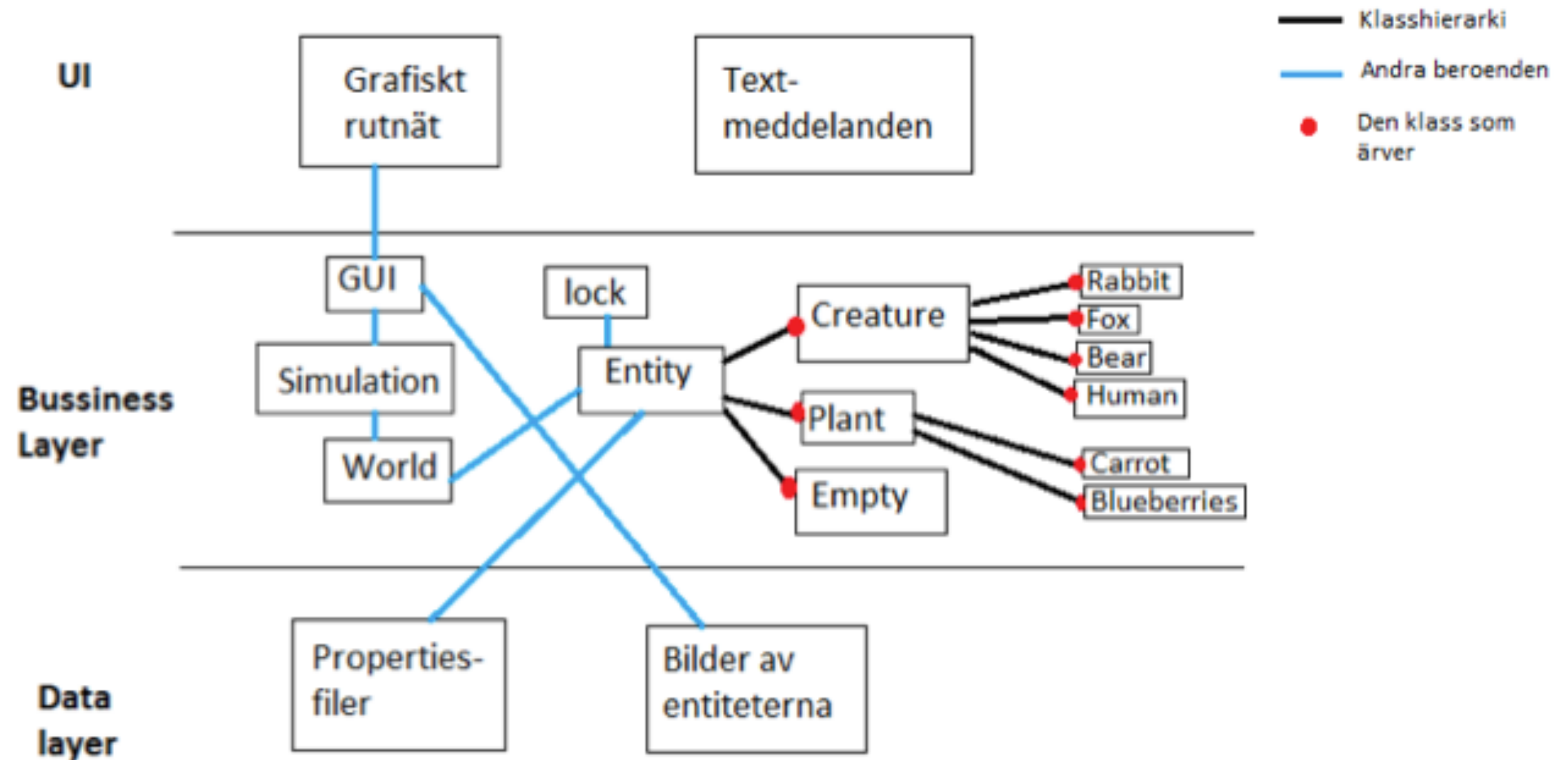


Fig 3. Systemarkitektur, översikt (orange: back-end (Erlang), blå: front-end (Python)).

A hybrid system using both Erlang and Python is described

System architecture example (3)



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.

Mer information om projektet läggs löpande upp på kursens hemsida



9 – Group project



Kom igång



Timeline



Bedömning

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.