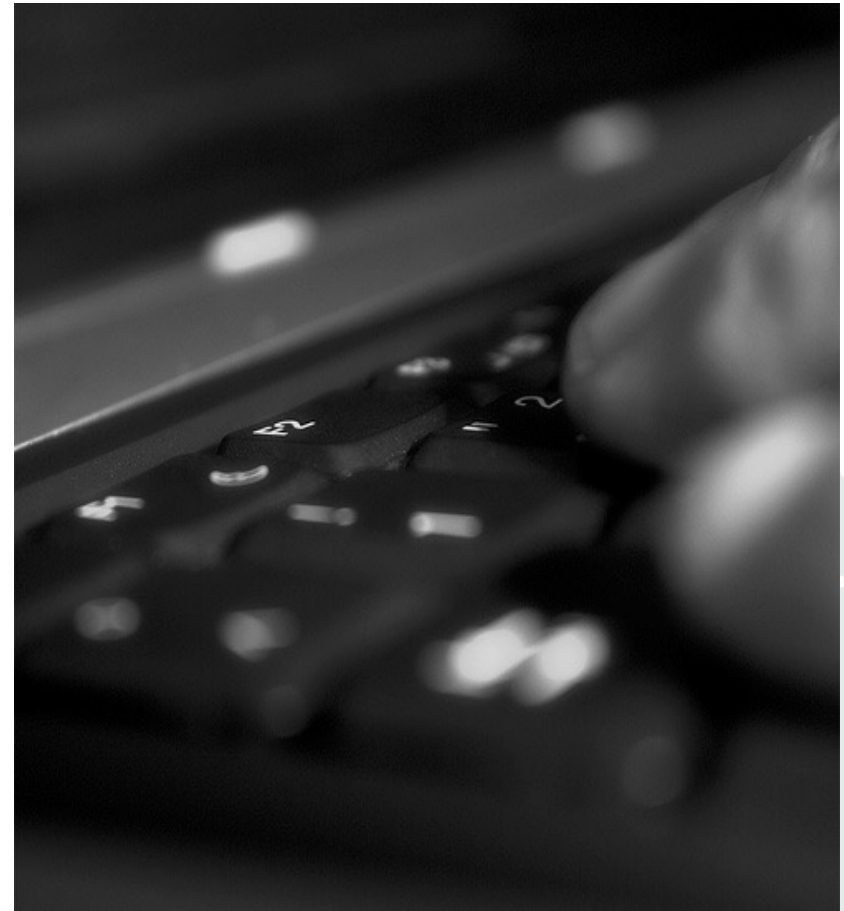


## Exercise 4 Business Informatics 2 (PWIN)

Management of IT Projects &  
Software Engineering

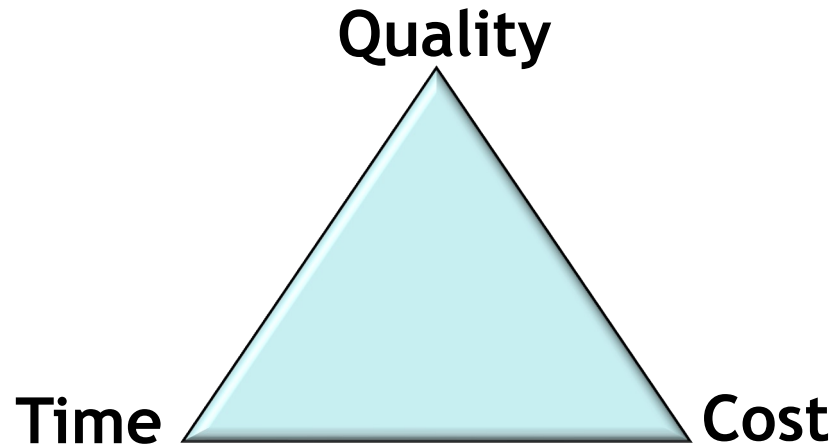
[www.m-chair.de](http://www.m-chair.de)



Jenser (Flickr.com)

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

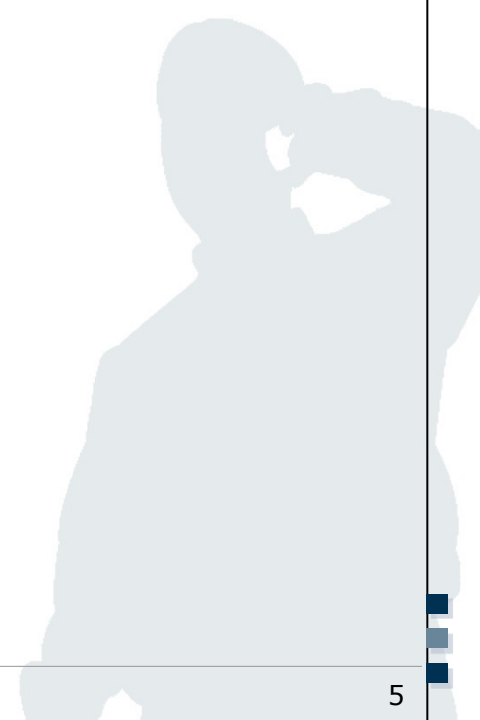
Explain what is meant by “magic triangle of project management”.



Requirement Change	Consequences	
Shorter time	Higher costs	Reduced quality or scope
Reduced costs	More time	Reduced quality or scope
Higher quality	More time	Higher costs

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

What are the “SMART” project objectives? Explain them at the example of the InstaMatch Service.



## Specific:

- Desired objectives should specify what should be achieved and include some quantitative targeted values for the end product.

## Measurable:

- You should be able to measure whether the objectives have been met or not.

## Attainable:

- The desired objective must be one that is actually feasible to achieve within the given time and cost parameters.

## Relevant:

- The desired objective should relate directly to the organisation's business needs and stated mission.

## Time-bound:

- The boundaries for completion date of the desired objective should be either a specific date or time.

## **S**pecific:

- Improve quality of the personal profile matching
- Reduce calculation time for the personal profile matching

## **M**easurable:

- Number of matching requests per performed date
- Reduce calculation time for the personal profile matching

## **A**ttainable:

- Five matching requests per performed date
- Reduce calculation time for the personal profile matching down to 3ms

## **R**elevant:

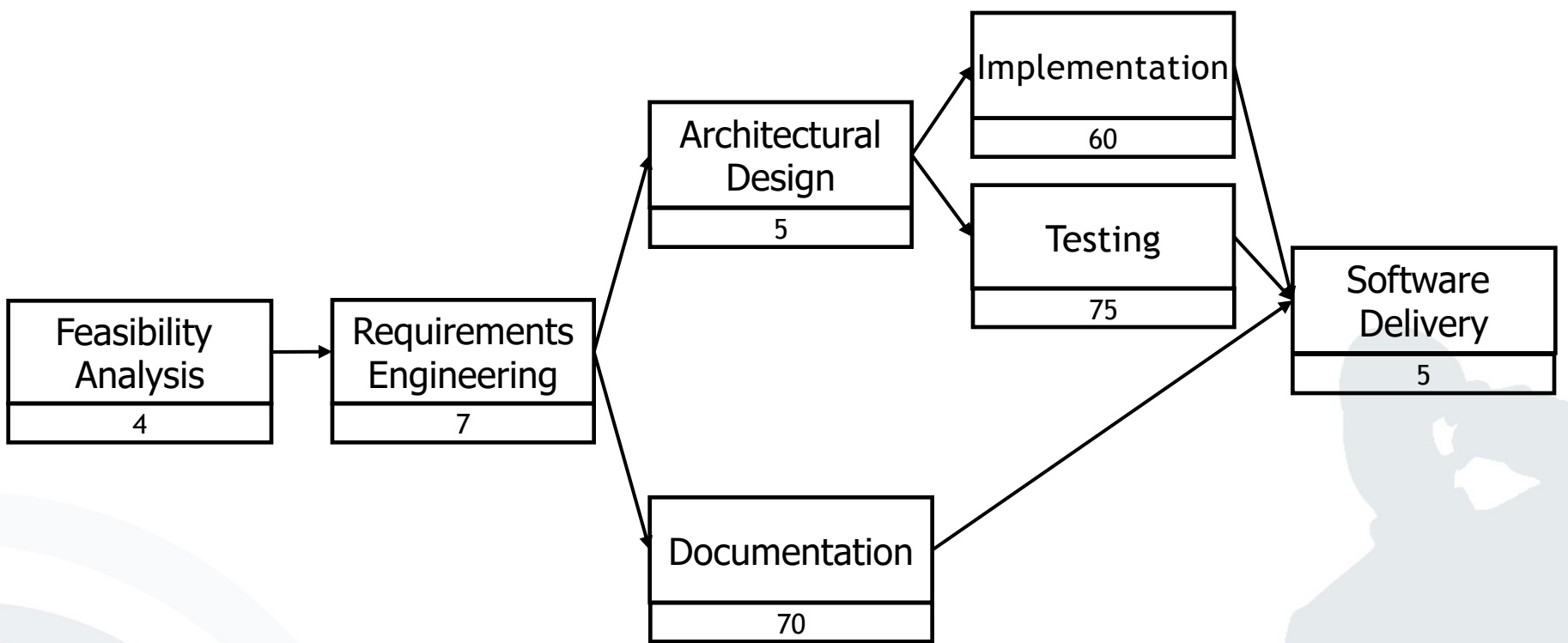
- Personal profile matching constitutes the core functionality of the InstaMatch system

## **T**ime-bound:

- Objective completion until end of 2020

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

# Excursion: Network Plan



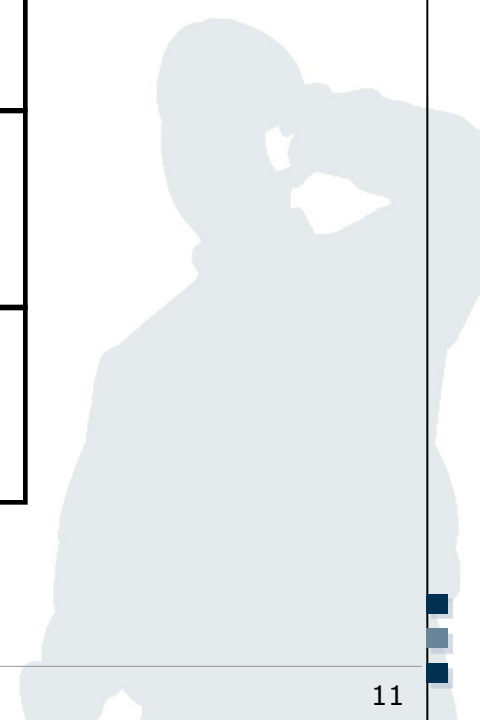
Order relation →

The project manager of a software company wants you to prepare a network plan for an upcoming software development project. His assistant has gathered the activity time estimates and their dependencies shown below.

Activity ID	Activity Description	Duration (days)	Preceding Activities
A	Feasibility Analysis	4	-
B	Requirements Engineering	7	{A}
C	Documentation	32	{B}
D	Hardware Procurement	3	{B}
E	Implementation	40	{D, F}
F	Architectural Design	5	{B}
G	Testing	28	{D, F}
H	Software Delivery	5	{C, E, G}

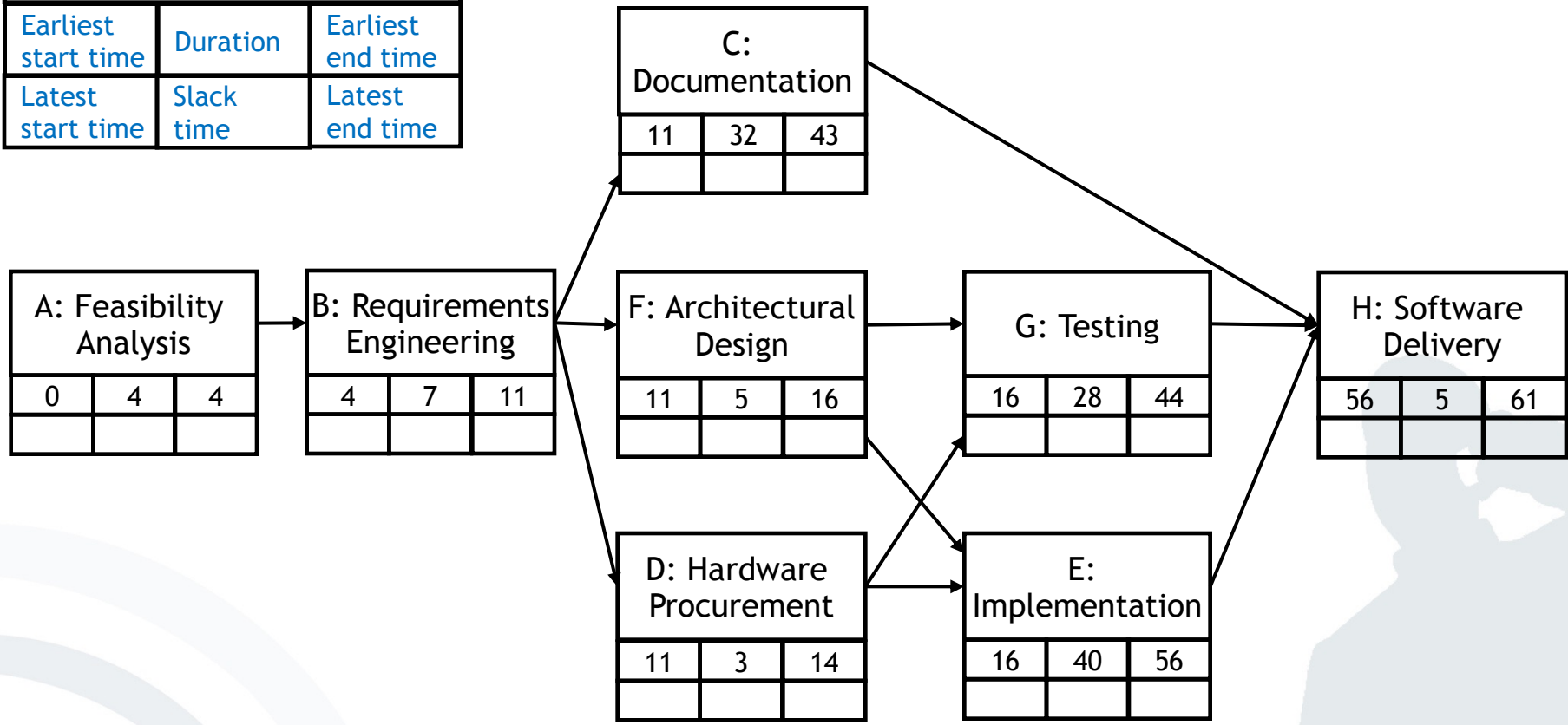
- a) Build a network plan according to the following schema and determine the critical path using activity slack calculations. Estimate how long the project will take.

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time



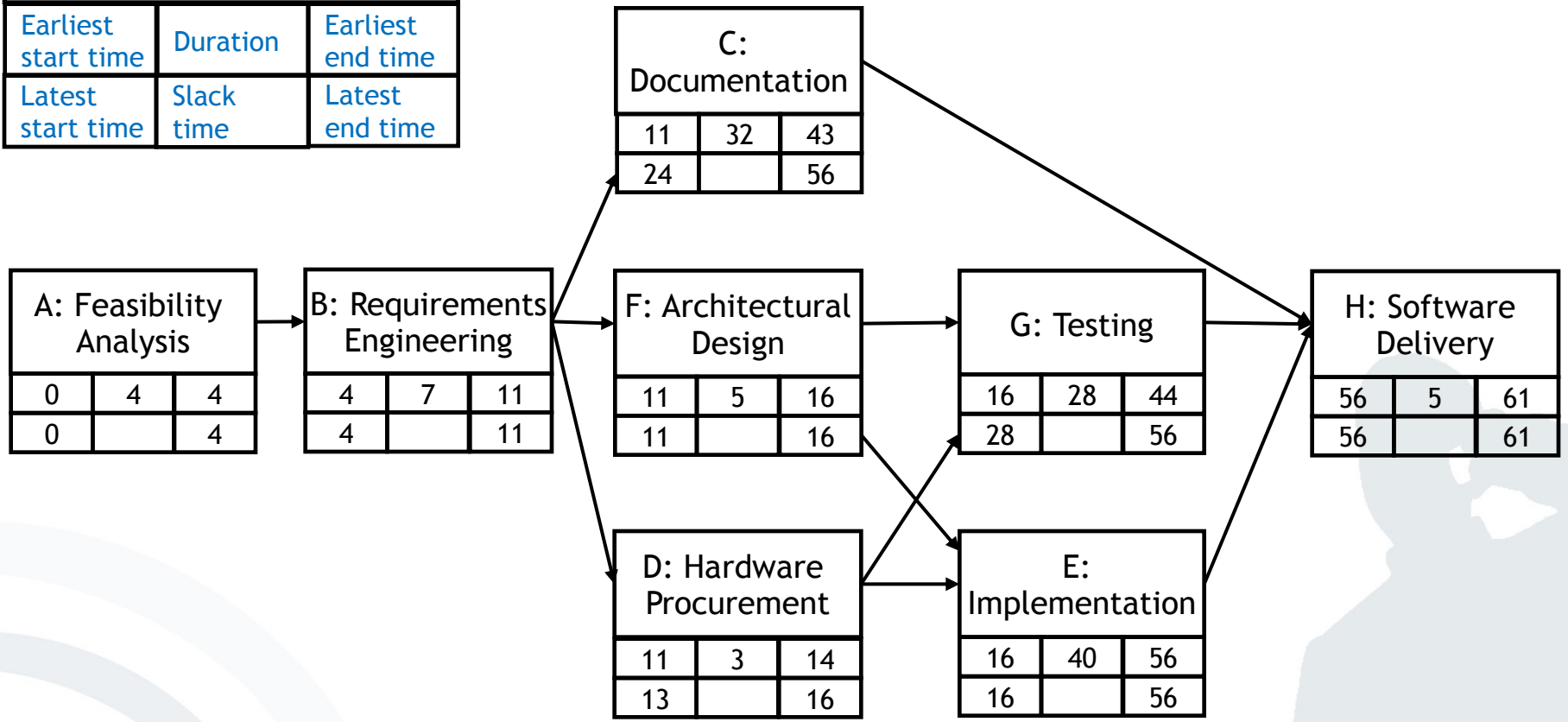
# Exercise 3a): Network Plan and Gantt Chart

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time



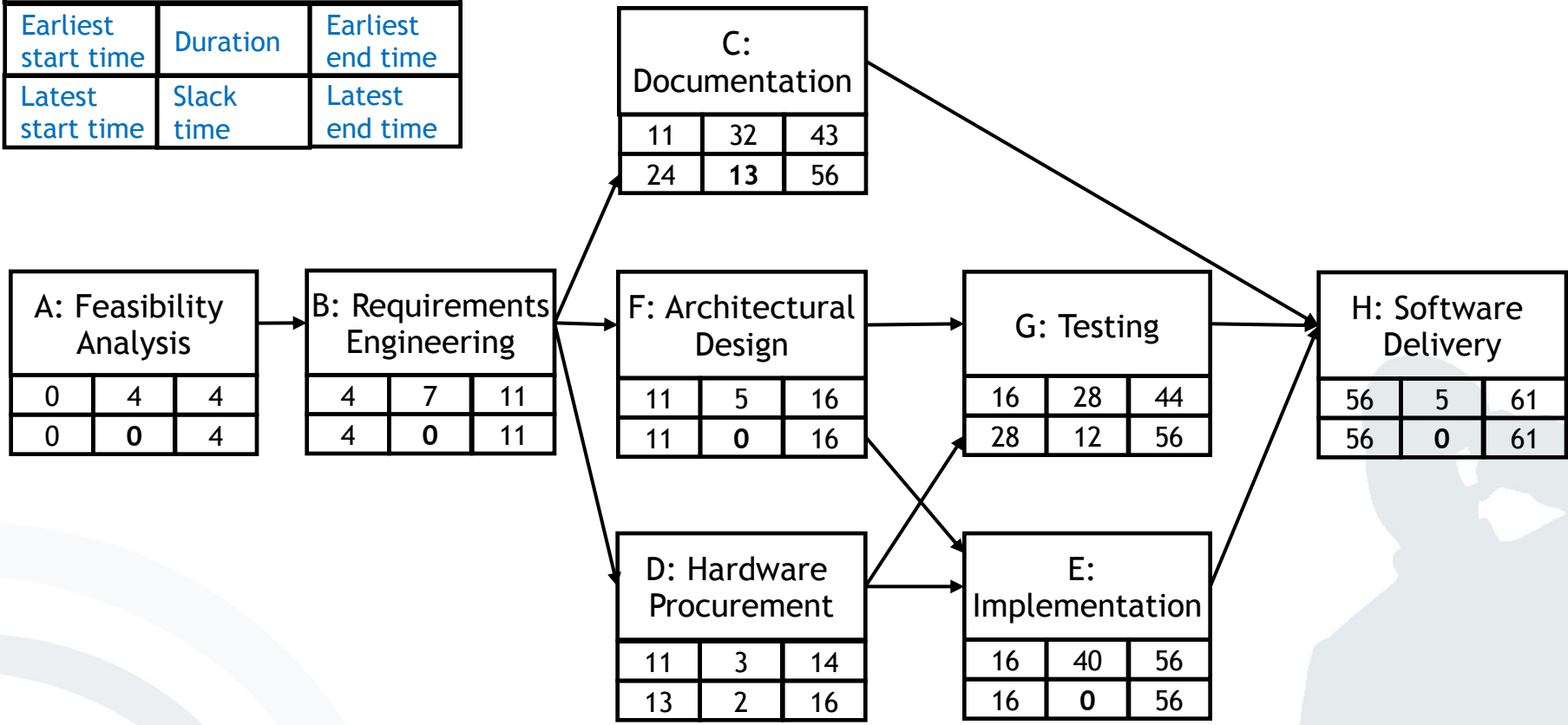
# Exercise 3a): Network Plan and Gantt Chart

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time



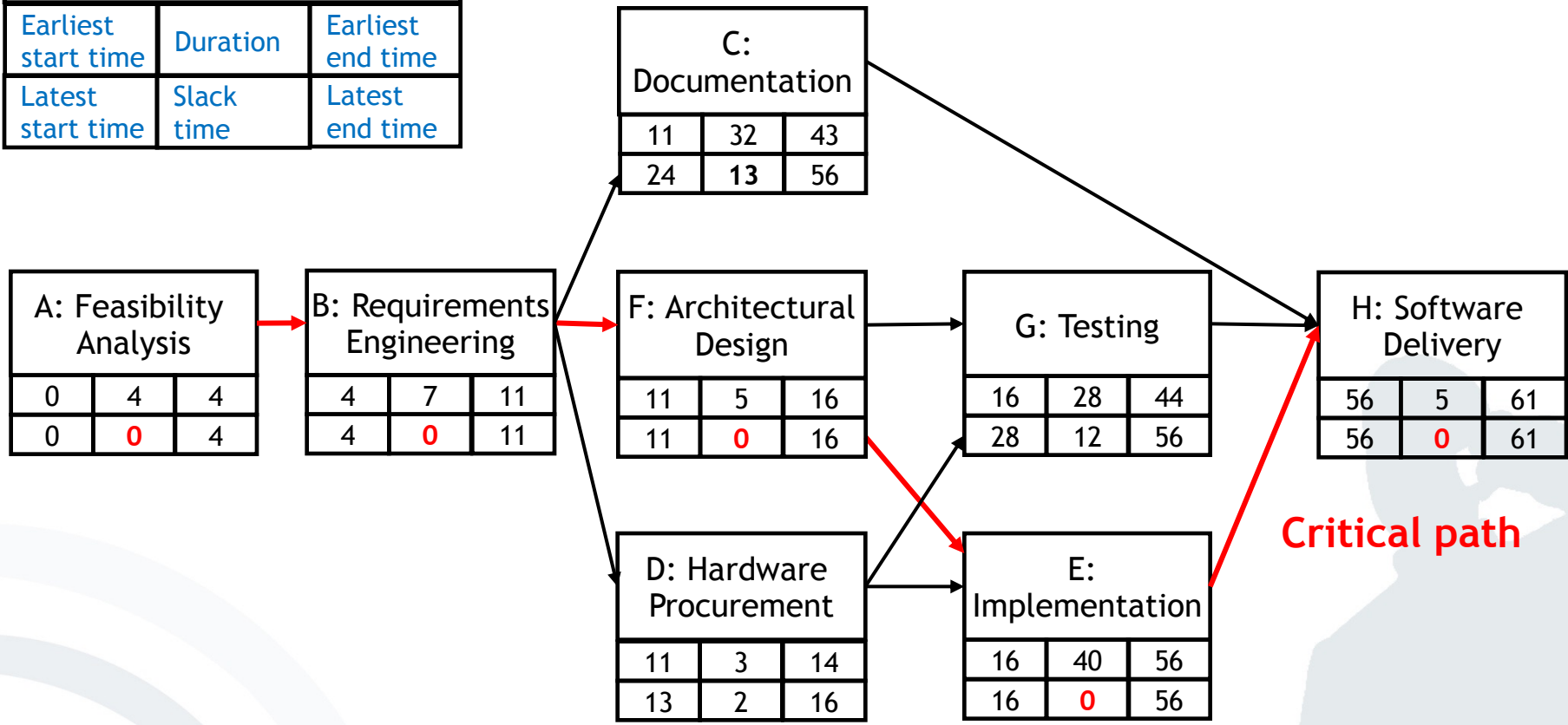
# Exercise 3a): Network Plan and Gantt Chart

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time



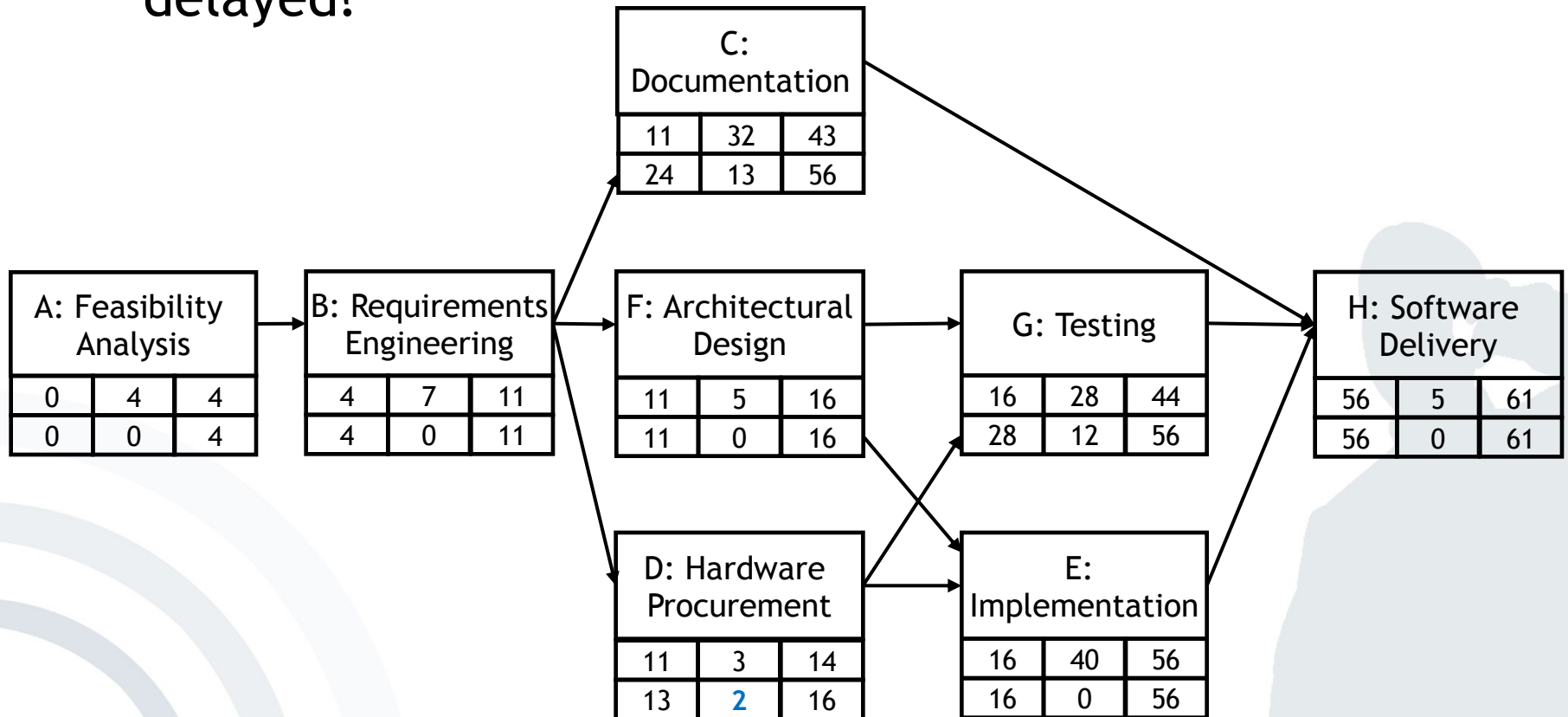
# Exercise 3a): Network Plan and Gantt Chart

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time



**Critical path**

b) How long can activity “D: Hardware Procurement” be delayed?

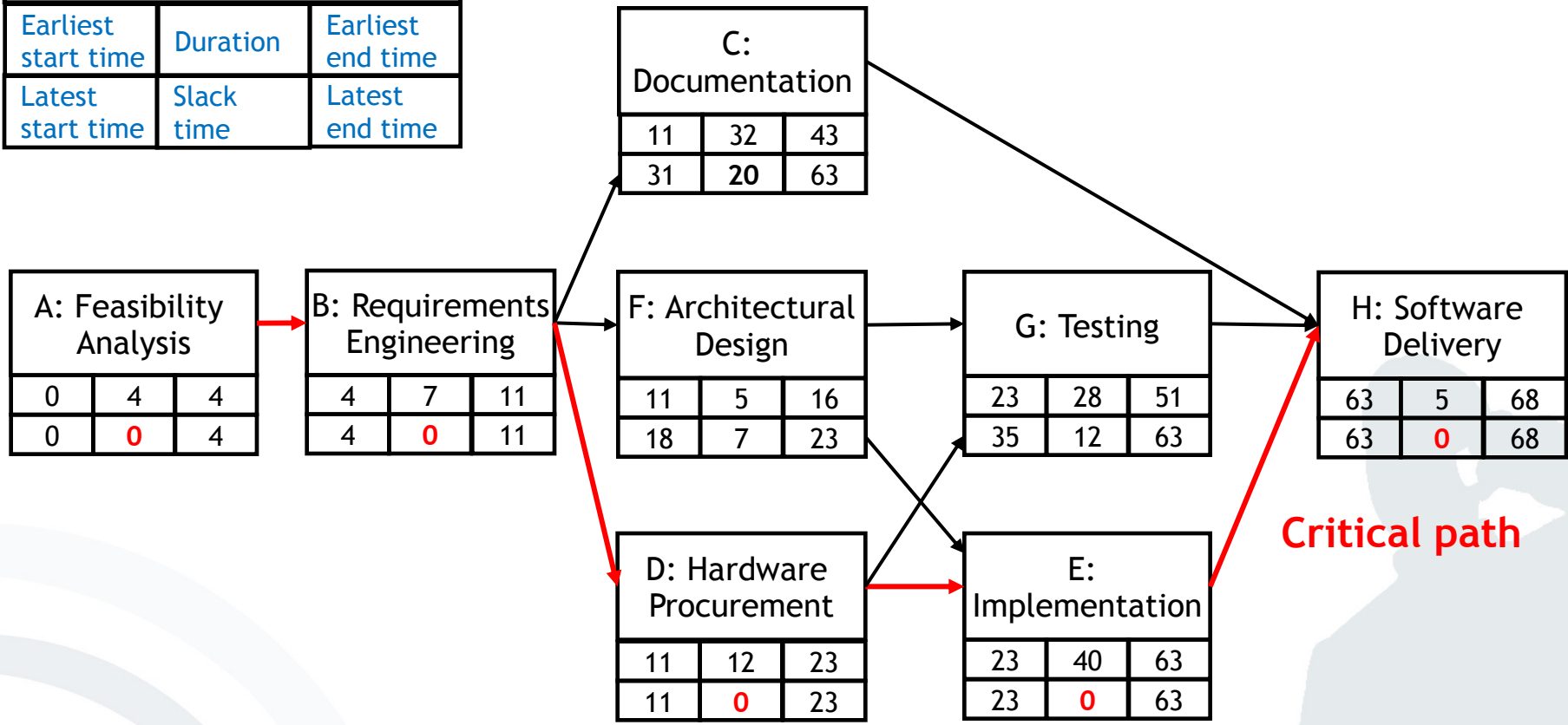


- c) Because of some internal problems with the procurement process, the project manager now expects activity D to require 12 days. Rebuild the network plan, highlight the critical path and describe the impact on the project schedule.

Activity ID	Activity Description	Duration (days)	Preceding Activities
A	Feasibility Analysis	4	-
B	Requirements Engineering	7	{A}
C	Documentation	32	{B}
D	Hardware Procurement	12	{B}
E	Implementation	40	{D, F}
F	Architectural Design	5	{B}
G	Testing	28	{D, F}
H	Software Delivery	5	{C, E, G}

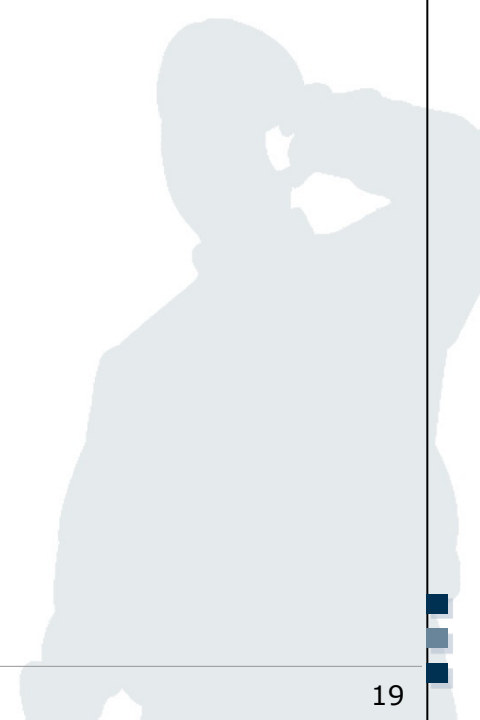
# Exercise 3c): Network Plan and Gantt Chart

Activity Description		
Earliest start time	Duration	Earliest end time
Latest start time	Slack time	Latest end time

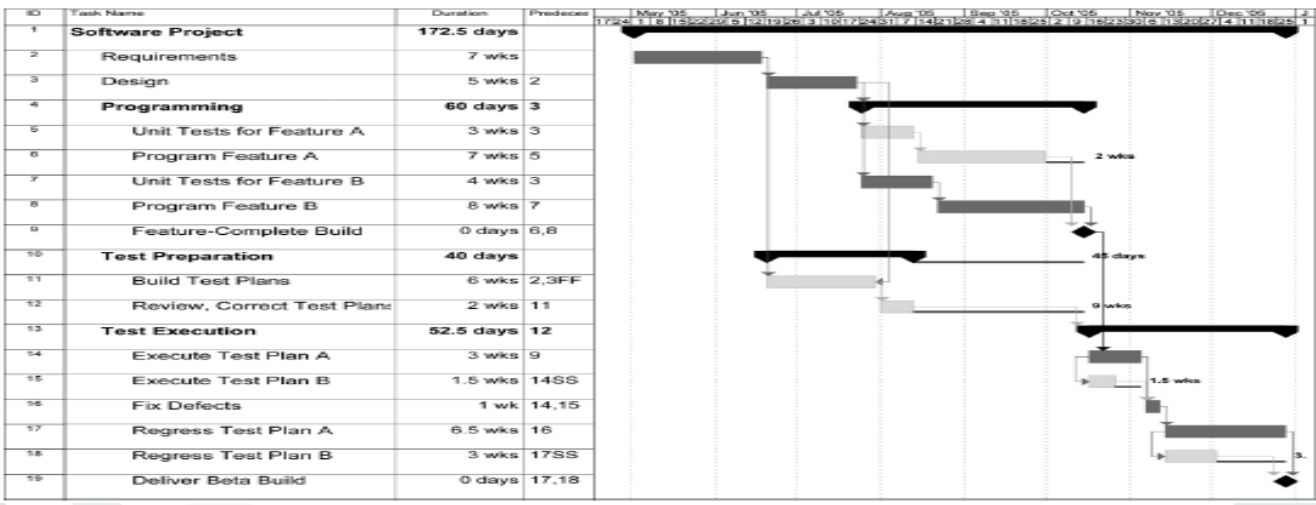
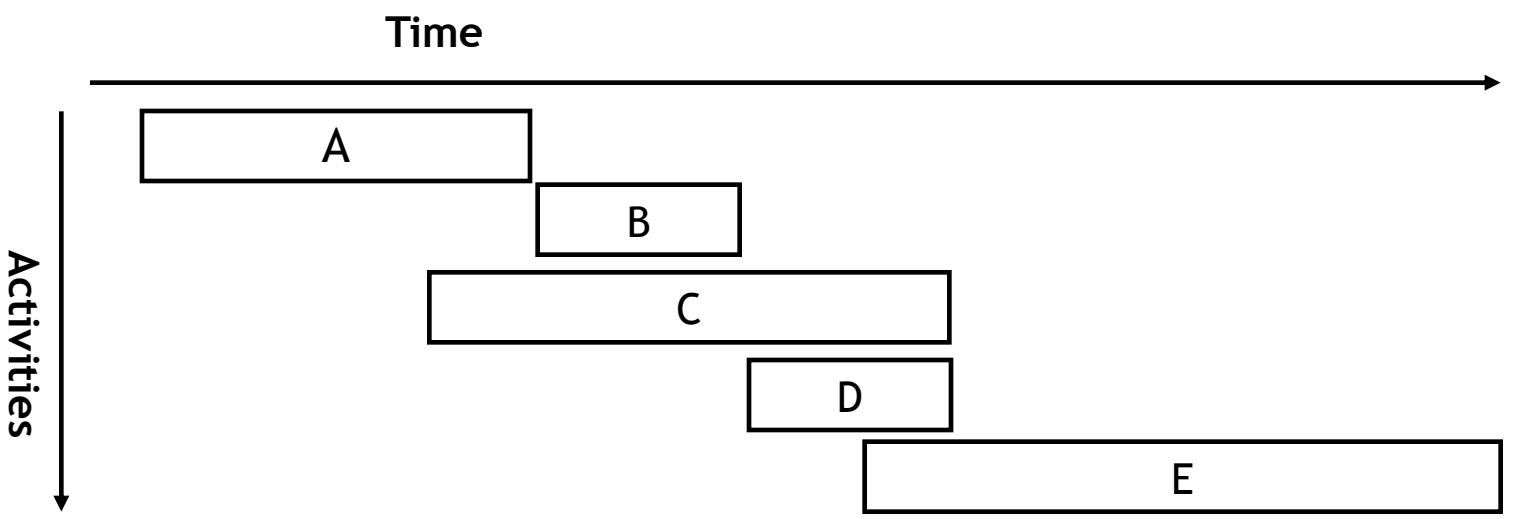


**Critical path**

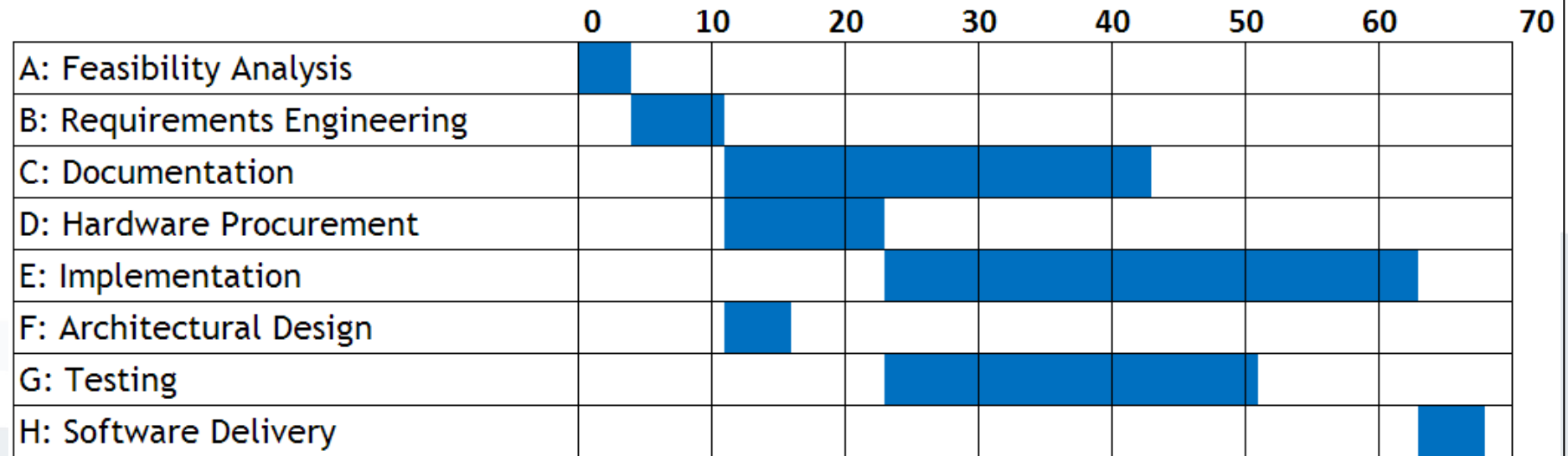
- d) The project manager now asks you to develop a Gantt chart based on the updated network plan.



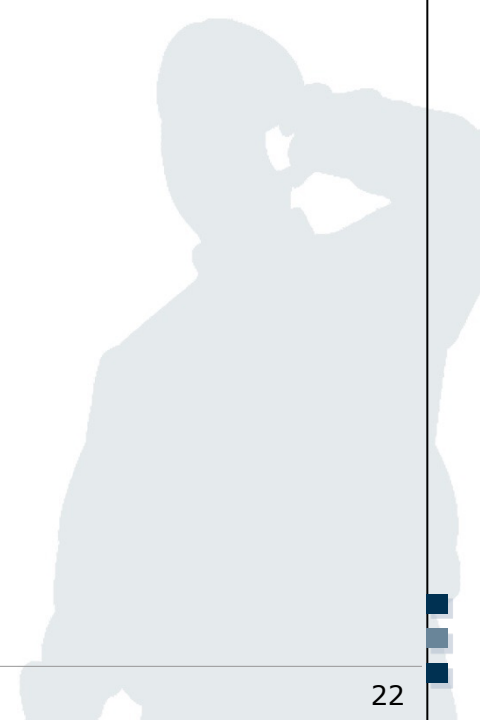
# Excursion: Gantt Chart



d) The project manager now asks you to develop a Gantt chart based on the updated network plan.

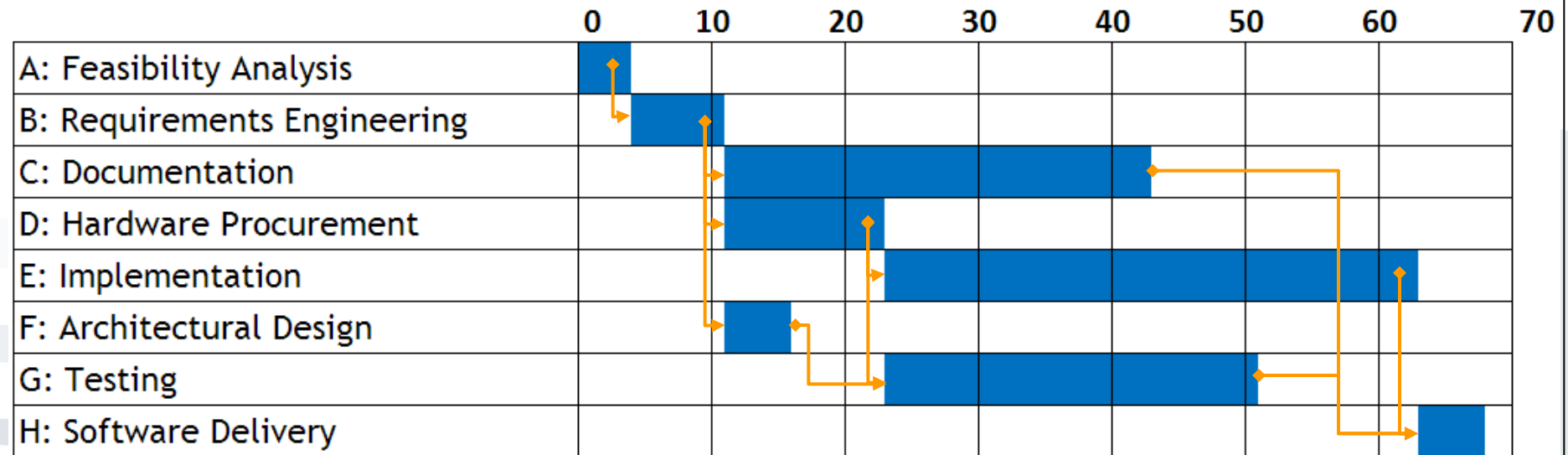


f) How could you improve the visualisation of Gantt charts?



f) How could you improve the visualisation of Gantt charts?

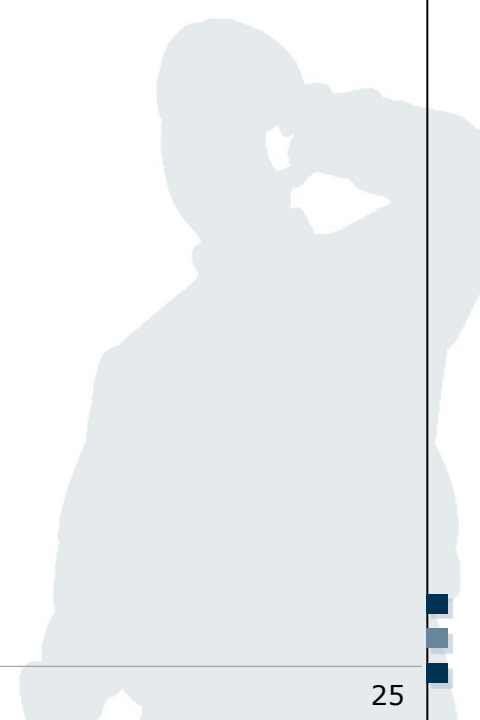
**Networked Gantt charts** visualize the dependencies between activities.



- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

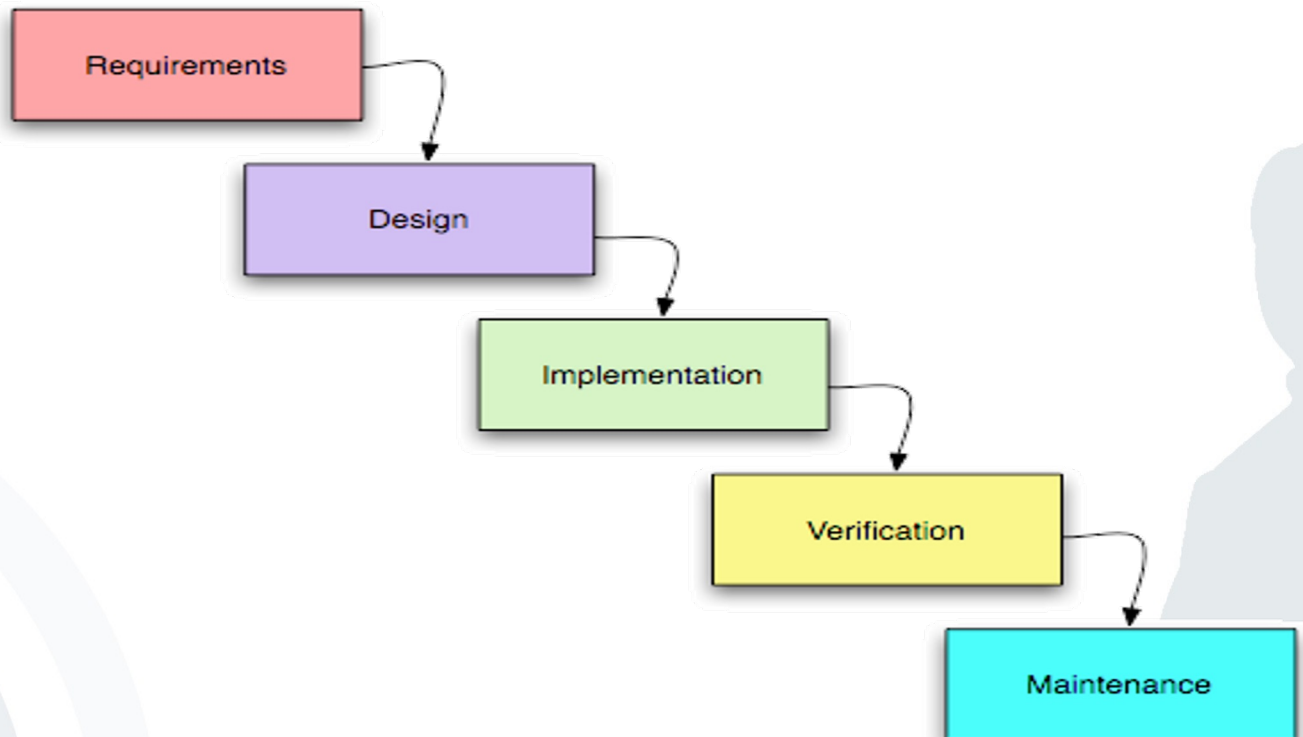
# Exercise 4: Software Development Process Models

- Name and describe three software development process models and select one of them for the InstaMatch Service. Motivate your choice.



## Waterfall model

- First described by Royce in 1970
- There seem to be at least as many versions as there are authorities - perhaps more



- One or more documents are produced after each phase and “signed off”.
- Points to note:
  - “Water does not flow up”.
    - it is difficult to change artifact produced in the previous phase.
  - This model should be used only when the requirements are well understood.
  - Reflects engineering practice.
  - Simple management model.

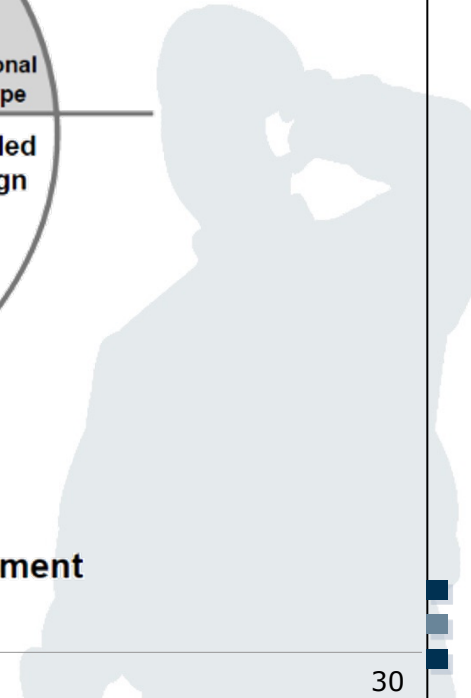
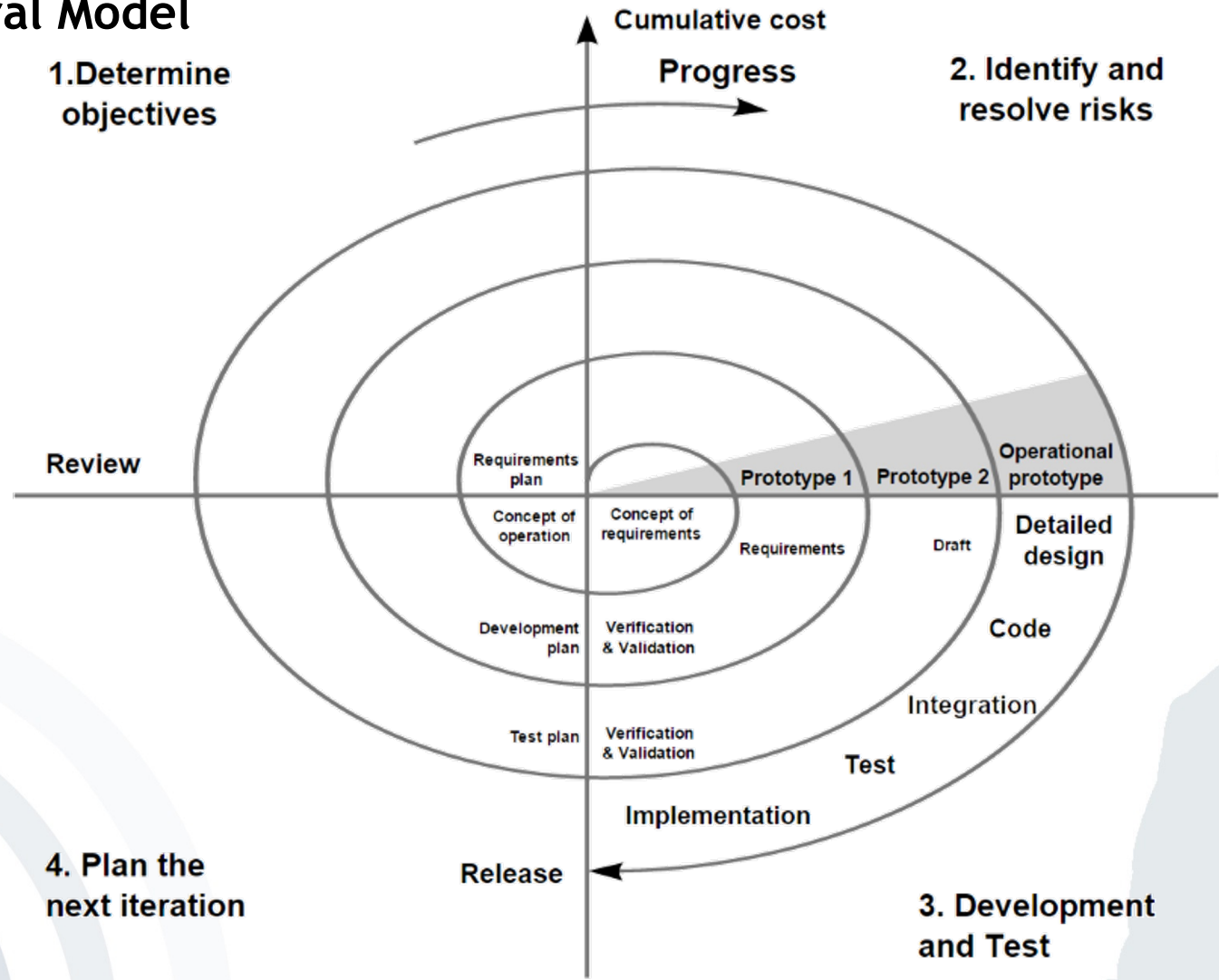


# Exercise 4: Software Development Process Models

- Similar to pure waterfall model but makes explicit the dependency between development and verification activities.
- The left half of the V represents *development* and the right half *system validation*.
- Note the requirements specification includes requirements elicitation and analysis.

# Exercise 4: Software Development Process Models

## Spiral Model



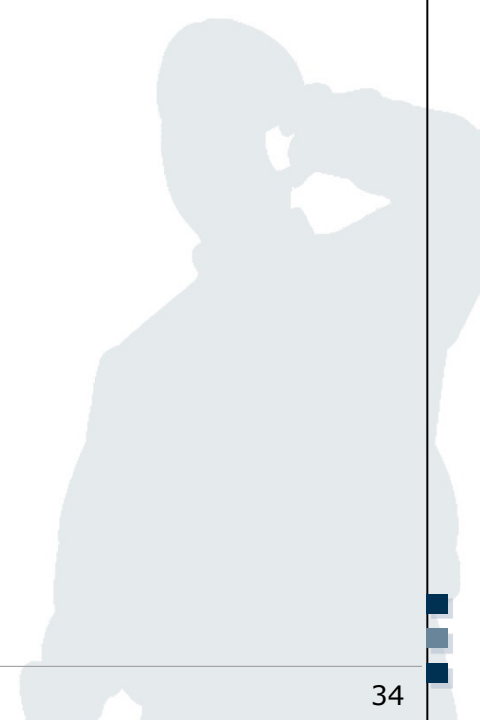
# Exercise 4: Software Development Process Models

- Basic Concept
  - Develop an initial implementation, demonstrate it to user, get feedback and refine it until an adequate system has been produced.
- Advantages
  - Estimates for budget, schedule, etc. become more realistic as work progresses
- Disadvantages
  - Requires expertise in risk evaluation and mitigation
  - Appropriate only for large systems

- Process model proposal for the InstaMatch®  
Service: V-Model
- Motivation
  - The InstaMatch® system is a complex system. The V-model was designed for complex systems.
  - The V-model makes explicit the dependency between development and validation and allows to jump back to earlier development phases.

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

5 a) What is meant by a well-formed XML document?



An XML document is **well-formed**, if:

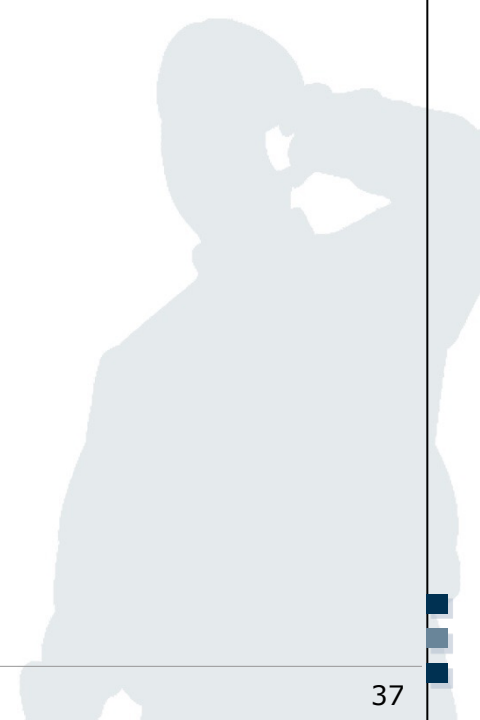
```
<?xml version="1.0"?>
<flirt>
  <name>Daisy</name>
  <mobile>436508469249</m
obile>
  <email>daisy@m-
chair.de</email>
  <city>Innsbruck</city>
  <first date>2020-01-
23</first date>
  <last date>2020-05-
01</last date>
  <birthday>1993-11-
13</birthday>
  <vegetarian>no</vegetar
ian>
  <status>single</status>
</flirt>
```

An XML document is **well-formed**, if:

- The element tags are case-sensitive
- No un-escaped special syntax characters:  
& ' < > "
- The element tags are correctly nested
- Only one "root" element
- Properly encoded legal unicode characters

```
<?xml version="1.0"?>
<flirt>
  <name>Daisy</name>
  <mobile>436508469249</m
obile>
  <email>daisy@m-
chair.de</email>
  <city>Innsbruck</city>
  <first date>2020-01-
23</first date>
  <last date>2020-05-
01</last date>
  <birthday>1993-11-
13</birthday>
  <vegetarian>no</vegetar
ian>
  <status>single</status>
</flirt>
```


5 b) Indicate which of the following XML documents are well-formed? Mark the mistakes and correct them.



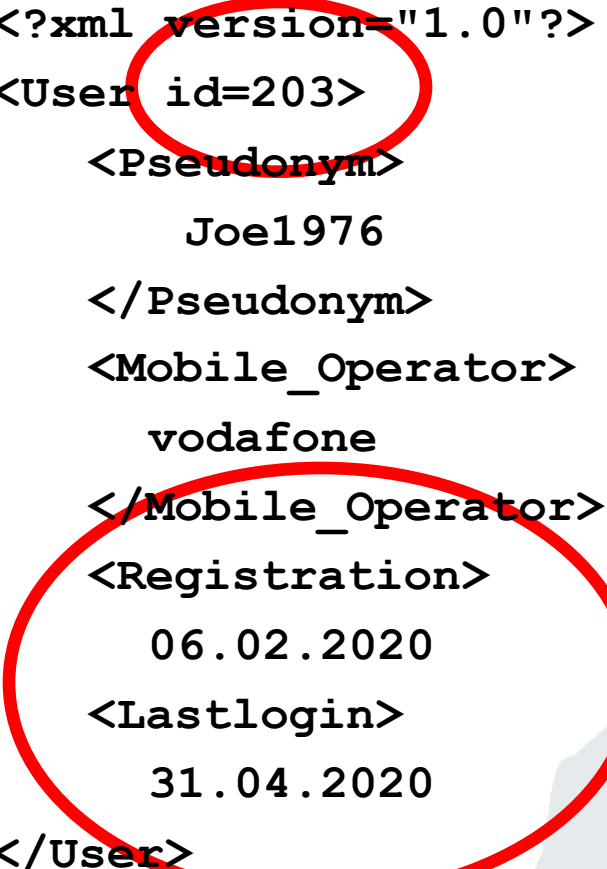
```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<?xml version="1.0"?>
<User id=203>
  <Pseudonym>
    Joe1976
  </Pseudonym>
  <Mobile_Operator>
    vodafone
  </Mobile_Operator>
  <Registration>
    06.02.2020
  <Lastlogin>
    31.04.2020
</User>
```

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```



```
<?xml version="1.0"?>
<User id=203>
  <Pseudonym>
    Joe1976
  </Pseudonym>
  <Mobile_Operator>
    vodafone
  </Mobile_Operator>
  <Registration>
    06.02.2020
  <Lastlogin>
    31.04.2020
  </User>
```



```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<?xml version="1.0"?>
<User id="203">
  <Pseudonym>
    Joe1976
  </Pseudonym>
  <Mobile_Operator>
    vodafone
  </Mobile_Operator>
  <Registration>
    06.02.2020
  </Registration>
  <Lastlogin>
    31.04.2020
  </Lastlogin>
</User>
```

```
<?xml version="1.0"?>
<Date>
  <Places>
    An der Hauptwache 7
  </Place>
  <Time>
    25.03.2020, 21:15-0:15
  </Time>
  <Meeting_Point>
    Starbucks
  <People>
    Gina
  </Meeting_Point>
  </People>
  <People>
    Jimmy
  </People>
  <Activity>
    Cocktails
  </Activity>
  <Comment>
    Spend Gina 2 Caipis
  </Comment>
</date>
```

```
<?xml version="1.0"?>
<Date>
  <Place>
    Theodor-W.-Adorno-Platz 5
  </Place>
  <Time>
    25.03.2020, 16:15-19:15
  </Time>
  <Meeting_Point>
    Sturm und Drang
  </Meeting_Point>
  <People>
    Joe1976
  </People>
  <People>
    Jenny23
  </People>
  <Activity>
    Drinking beer
  </Activity>
  <Comment>
    Joe1976 wears sun glasses!
  </Comment>
</Date>
```

```

<?xml version="1.0"?>
<Date>
  <Places>
    An der Hauptwache 7
  </Place>
  <Time>
    25.03.2020, 21:15-0:15
  </Time>
  <Meeting_Point>
    Starbucks
  <People>
    Gina
  </Meeting_Point>
  </People>
  <People>
    Jimmy
  </People>
  <Activity>
    Cocktails
  </Activity>
  <Comment>
    Spend Gina 2 Caipis
  </Comment>
</date>

```

```

<?xml version="1.0"?>
<Date>
  <Place>
    Theodor-W.-Adorno-Platz 5
  </Place>
  <Time>
    25.03.2020, 16:15-19:15
  </Time>
  <Meeting_Point>
    Sturm und Drang
  </Meeting_Point>
  <People>
    Joe1976
  </People>
  <People>
    Jenny23
  </People>
  <Activity>
    Drinking beer
  </Activity>
  <Comment>
    Joe1976 wears sun glasses!
  </Comment>
</Date>

```



```

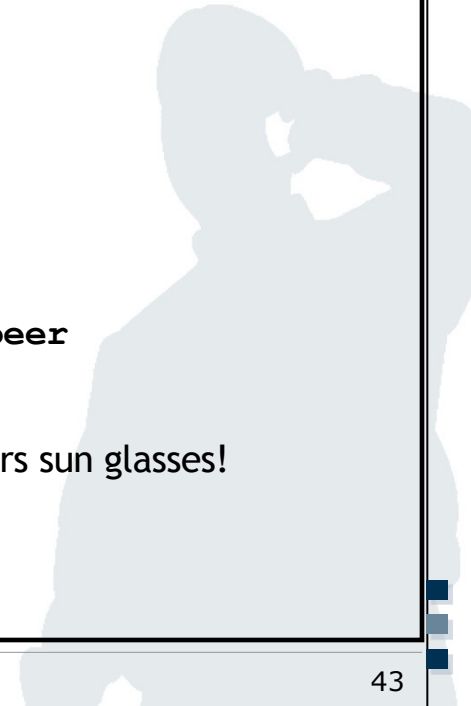
<?xml version="1.0"?>
<Date>
  <Places>
    An der Hauptwache 7
  </Place>
  <Time>
    25.03.2020, 21:15-0:15
  </Time>
  <Meeting_Point>
    Starbucks
  </Meeting_Point>
  <People>
    Gina
  </People>
  <People>
    Jimmy
  </People>
  <Activity>
    Cocktails
  </Activity>
  <Comment>
    Spend Gina 2 Caipis
  </Comment>
</Date>

```

```

<?xml version="1.0"?>
<Date>
  <Place>
    Theodor-W.-Adorno-Platz 5
  </Place>
  <Time>
    25.03.2020, 16:15-19:15
  </Time>
  <Meeting_Point>
    Sturm und Drang
  </Meeting_Point>
  <People>
    Joe1976
  </People>
  <People>
    Jenny23
  </People>
  <Activity>
    Drinking beer
  </Activity>
  <Comment>
    Joe1976 wears sun glasses!
  </Comment>
</Date>

```



5 c) Explain why the following XML document is not well-formed. Correct the syntax error.

```
<?xml version="1.0"?>
<User id="203">
  <Sex>
    Male
  </Sex>
  <Age>
    21
  </Age>
</User>
<User id="194">
  <Sex>
    Female
  </Sex>
  <Age>
    23
  </Age>
</User>
```

Two root elements are defined.  
But well-formed XML documents  
only allow one root element.

```
<?xml version="1.0"?>
```

```
<Users>
```

```
  <User id="203">
```

```
    <Sex>
```

```
      Male
```

```
    </Sex>
```

```
    <Age>
```

```
      21
```

```
    </Age>
```

```
  </User>
```

```
  <User id="194">
```

```
    <Sex>
```

```
      Female
```

```
    </Sex>
```

```
    <Age>
```

```
      23
```

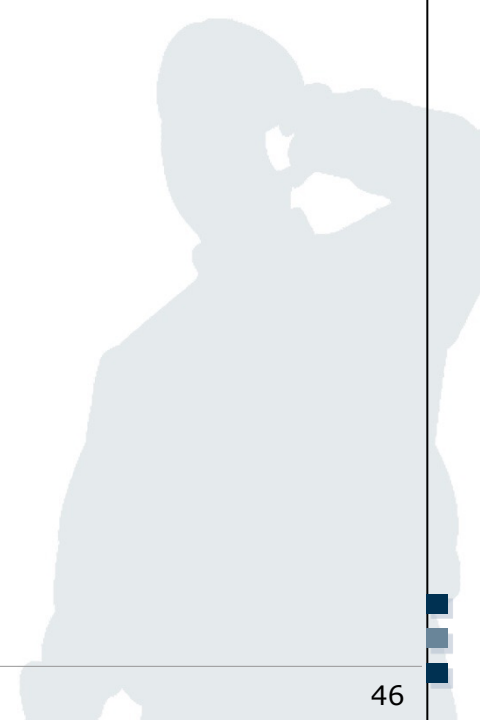
```
    </Age>
```

```
  </User>
```

```
</Users>
```

5 d) Create an XML document representing your two favorite dating locations.

Use at least two different tags for describing the locations and at least one attribute.



```
<?xml version="1.0"?>
```

```
<?xml version="1.0"?>
```

```
<Locations>
```

```
</Locations>
```

```
<?xml version="1.0"?>  
<Locations>  
  <Location id="332">  
  
  </Location>  
  <Location id="143">  
  
  </Location>  
</Locations>
```

```
<?xml version="1.0"?>
<Locations>
  <Location id="332">
    <Name>DASEIN</Name>
    <Rating>Good</Rating>
  </Location>
  <Location id="143">
    <Name>Sturm und Drang</Name>
    <Rating>Good</Rating>
  </Location>
</Locations>
```

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

Create a DTD for the XML document from Exercise 5 b).

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
] >
```



```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
<!ELEMENT User
(Pseudonym,Mobile_Operator,
Registration,Lastlogin)>
]>
```



```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
<!ELEMENT User
(Pseudonym,Mobile_Operator,
Registration,Lastlogin)>
<!ELEMENT Pseudonym (#PCDATA)>
]>
```



```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
<!ELEMENT User
(Pseudonym,Mobile_Operator,
Registration,Lastlogin)>

<!ELEMENT Pseudonym (#PCDATA)>
<!ELEMENT Mobile_Operator (#PCDATA)>

]>
```

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
  <!ELEMENT User
    (Pseudonym,Mobile_Operator,
    Registration,Lastlogin)>
  <!ELEMENT Pseudonym (#PCDATA)>
  <!ELEMENT Mobile_Operator (#PCDATA)>
  <!ELEMENT Registration(#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<User id="194">
  <Pseudonym>
    Jenny23
  </Pseudonym>
  <Mobile_Operator>
    t-mobile
  </Mobile_Operator>
  <Registration>
    03.02.2020
  </Registration>
  <Lastlogin>
    29.04.2020
  </Lastlogin>
</User>
```

```
<!DOCTYPE User [
  <!ELEMENT User
    (Pseudonym,Mobile_Operator,
    Registration,Lastlogin)>
  <!ELEMENT Pseudonym (#PCDATA)>
  <!ELEMENT Mobile_Operator (#PCDATA)>
  <!ELEMENT Registration(#PCDATA)>
  <!ELEMENT Lastlogin(#PCDATA)>
]>
```

Validate and correct the following XML document against the corresponding DTD document. Assume the given DTD is correct.

```
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
  </Address>
  <City>Berlin</City>
  <Age>23</Age>
  <Gender>male</male>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
  <Event>Networking</Event>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
  </Address>
  <City>Berlin</City>
  <Age>23</Age>
  <Gender>male</male>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
  <Event>Networking</Event>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
  </Address>
  <City>Berlin</City>
  <Age>23</Age>
  <Gender>male</male>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
  <Event>Networking</Event>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
Address, Age, Gender,
Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
  </Address>
  <City>Berlin</City>
  <Age>23</Age>
  <Gender>male</male>
  <Course>Scientific
  Work</Course>
  <Course>Economics</Course>
  <Event>Networking</Event>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
Address, Age, Gender,
Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
Address, Age, Gender,
Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
Address, Age, Gender,
Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
  </Street>
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
    Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
    Address, Age, Gender,
    Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
    ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    </Street>
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
  Work</Course>
  <Course>Economics</Course>
</Student>
```

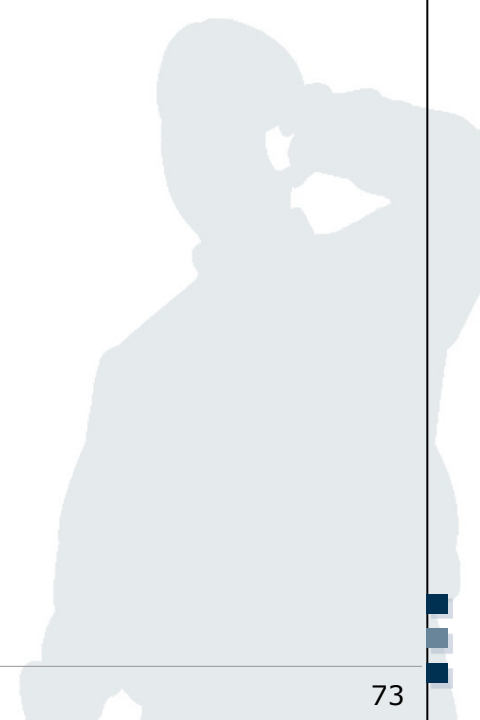
```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
  Address, Age, Gender,
  Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
  ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

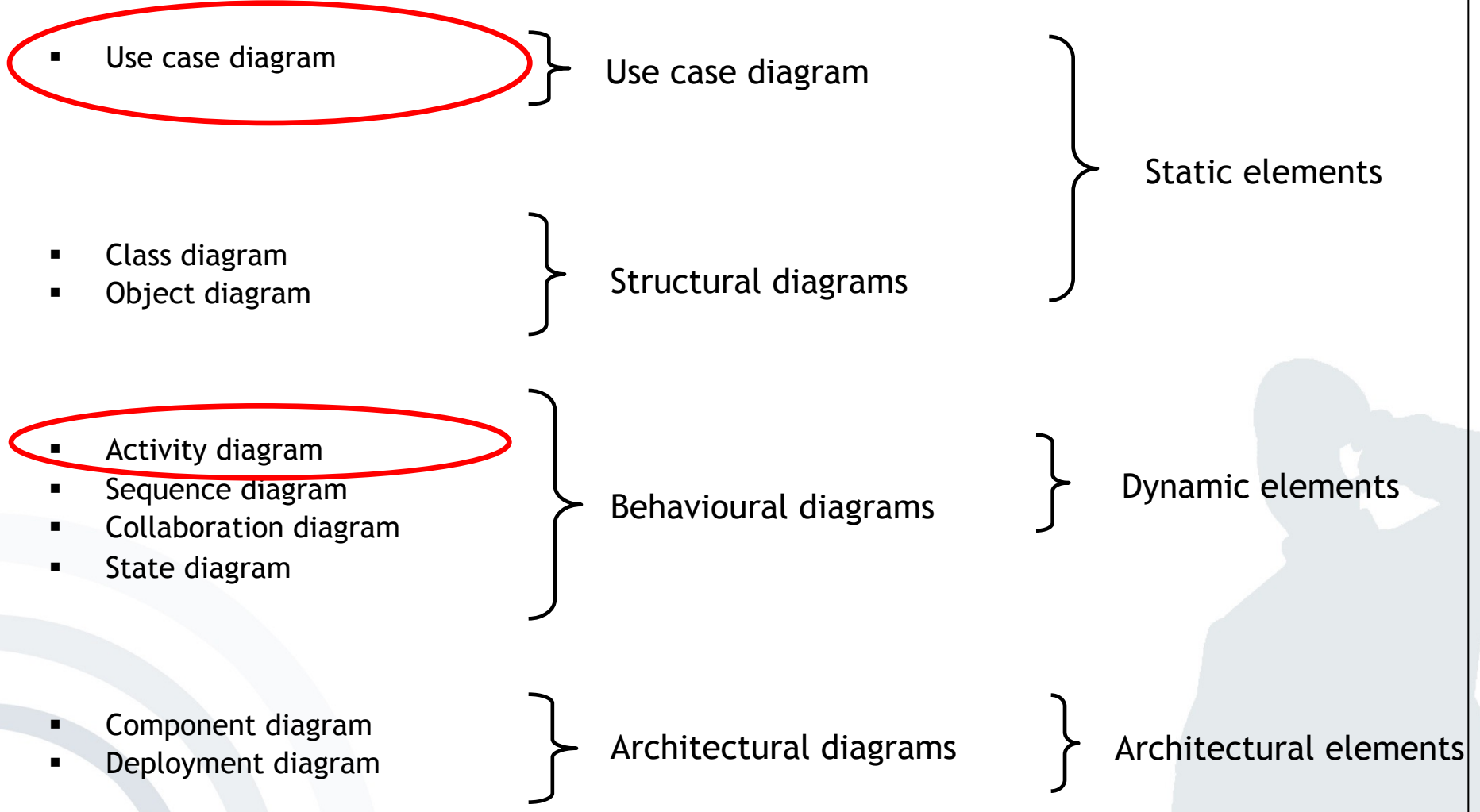
```
<?xml version="1.0"?>
<Student>
  <Name>Guenther M</Name>
  <Address>
    <Street>Cluestreet
    </Street>
    <ZipCode>63743</ZipCode>
    <City>Berlin</City>
  </Address>
  <Age>23</Age>
  <Gender>male</Gender>
  <Course>Scientific
  Work</Course>
  <Course>Economics</Course>
</Student>
```

```
<!DOCTYPE Student [
  <!ELEMENT Student (Name,
  Address, Age, Gender,
  Course*, Subject*)>
  <!ELEMENT Name (#PCDATA)>
  <!ELEMENT Address (Street,
  ZipCode, City)>
  <!ELEMENT Street (#PCDATA)>
  <!ELEMENT ZipCode (#PCDATA)>
  <!ELEMENT City (#PCDATA)>
  <!ELEMENT Age (#PCDATA)>
  <!ELEMENT Gender (#PCDATA)>
  <!ELEMENT Course (#PCDATA)>
  <!ELEMENT Subject (#PCDATA)>
]>
```

- Exercise 1: IT Project Management
- Exercise 2: IT Project Management
- Exercise 3: Network Plan and Gantt Chart
- Exercise 4: Software Development Process Models
- Exercise 5: Well-formed XML Documents
- Exercise 6: Document Type Definition
- Exercise 7: Unified Modeling Language

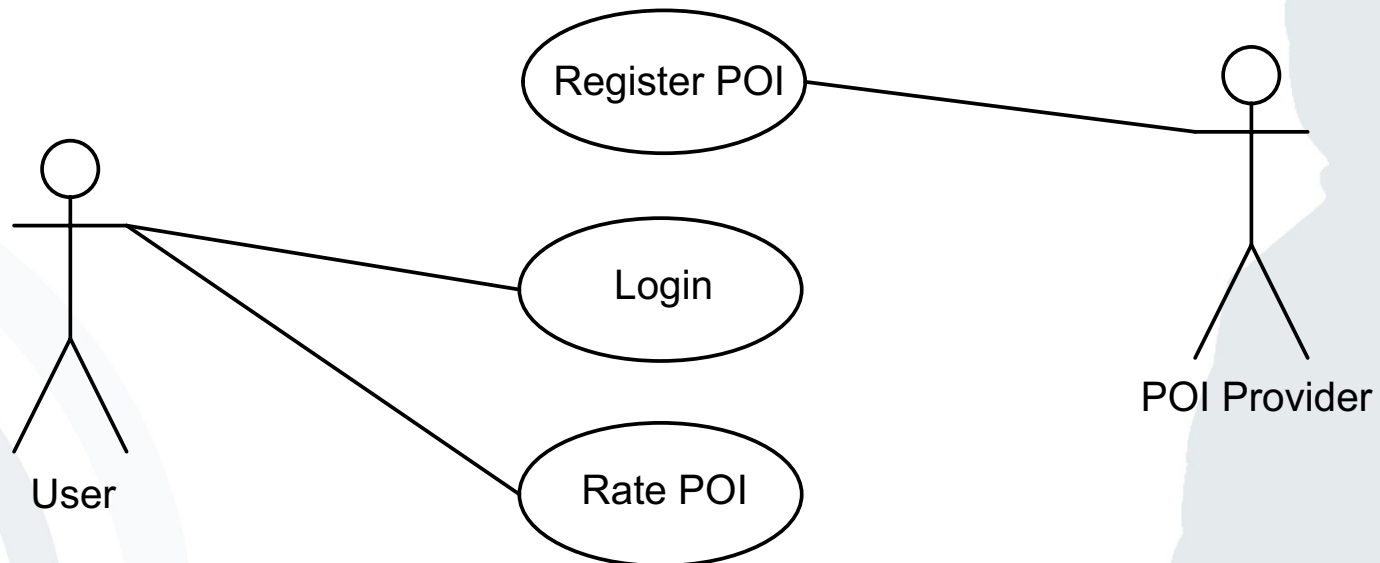
3 a) What are the differences between use case and activity diagrams?





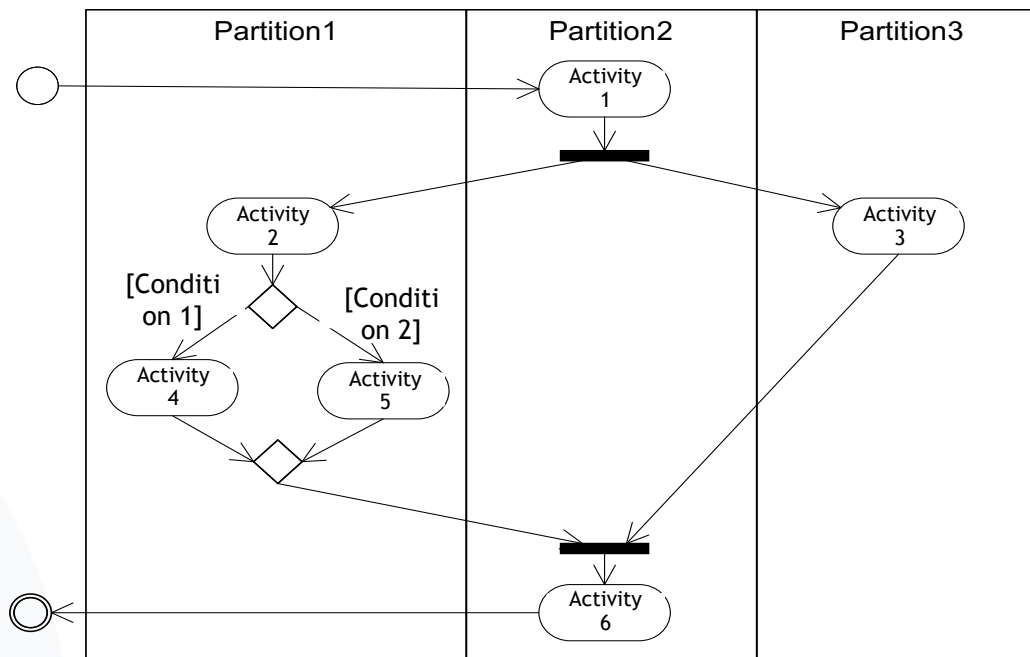
### Use case diagrams

- Use cases describe the functionality, which a system has to provide
- The sum of all “use cases” comprises the technical requirements of a system.
- Use cases define the interfaces between a user and the system
- Specification is developed together with the client/customer

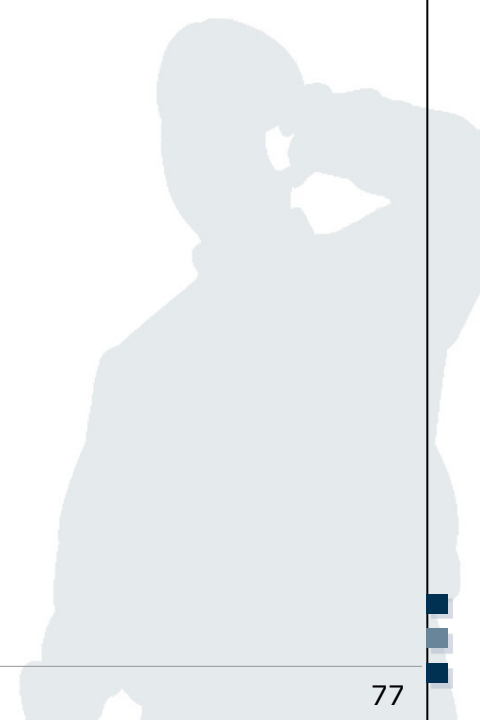


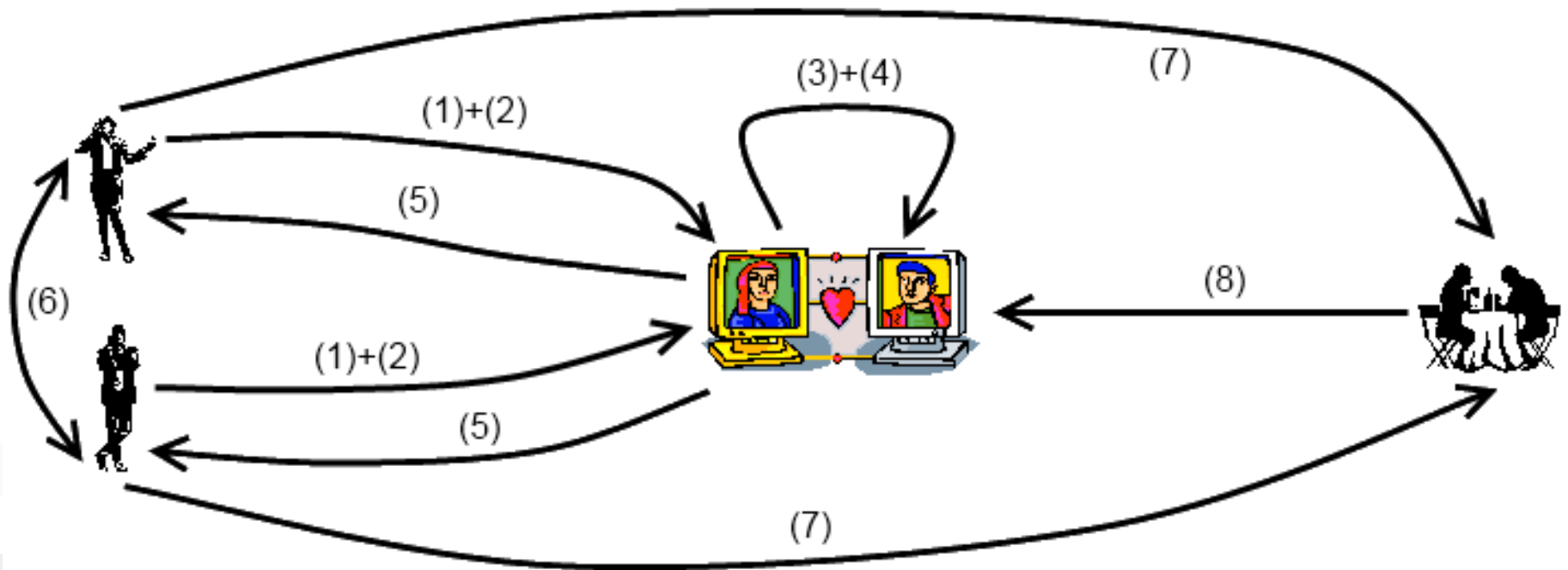
## Activity diagrams

- Activity diagrams are used to model workflows in a system.
- Central element: Activity  
An activity is an “action” within a process.
- Activities are structured by responsibilities.

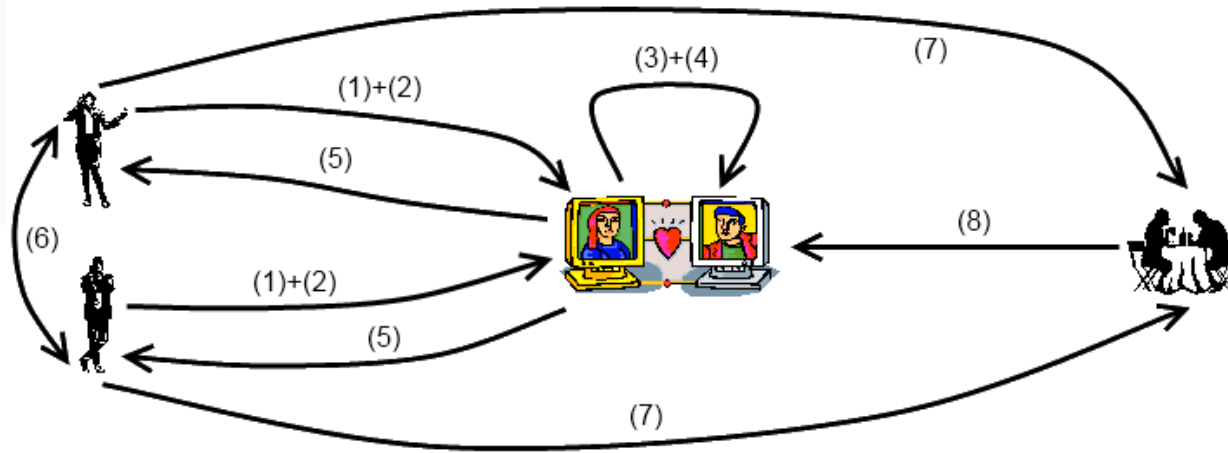


4 b) Develop a **use case** and an **activity diagram** for the InstaMatch<sup>®</sup> service based on Figure 1 of the InstaMatch scenario.



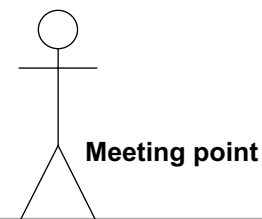
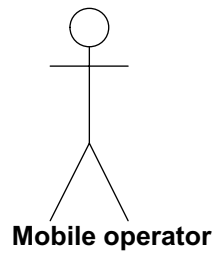


## Exercise 7 b): Unified Modeling Language



1. **Users register at InstaMatch®.** Thereby, they receive their pseudonyms and submit their personal profile information. Subsequently, InstaMatch® attempts to certify the profile attributes of the users.
2. In order to start searching for a date, users have to **activate the InstaMatch®** app on their mobile device.
3. InstaMatch® attempts to **find other InstaMatch® users in close proximity** who are also currently looking for a date.
4. **InstaMatch® matches the personal profiles** of all users in close proximity with each other.
5. If there is a match, **InstaMatch® informs the corresponding users** by presenting them with a list of matching pseudonyms.
6. InstaMatch® enables matching users to **communicate with each other** using text messages, chat or voice.
7. If users want to arrange a meeting, InstaMatch® **suggests a list of appropriate meeting points** based on the personal interests of the corresponding users as well as their current geographic location. Then, InstaMatch® **navigates the users to their meeting point.**
8. After the date, the **users can rate their date** on the InstaMatch® app. This rating is used to improve their next date matching process.

Use case diagram



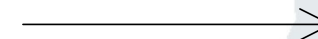
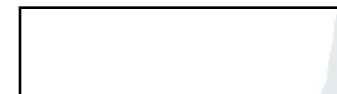
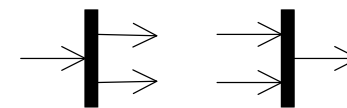
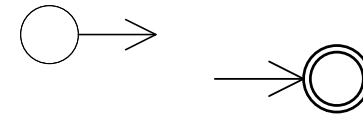
### Use case diagram



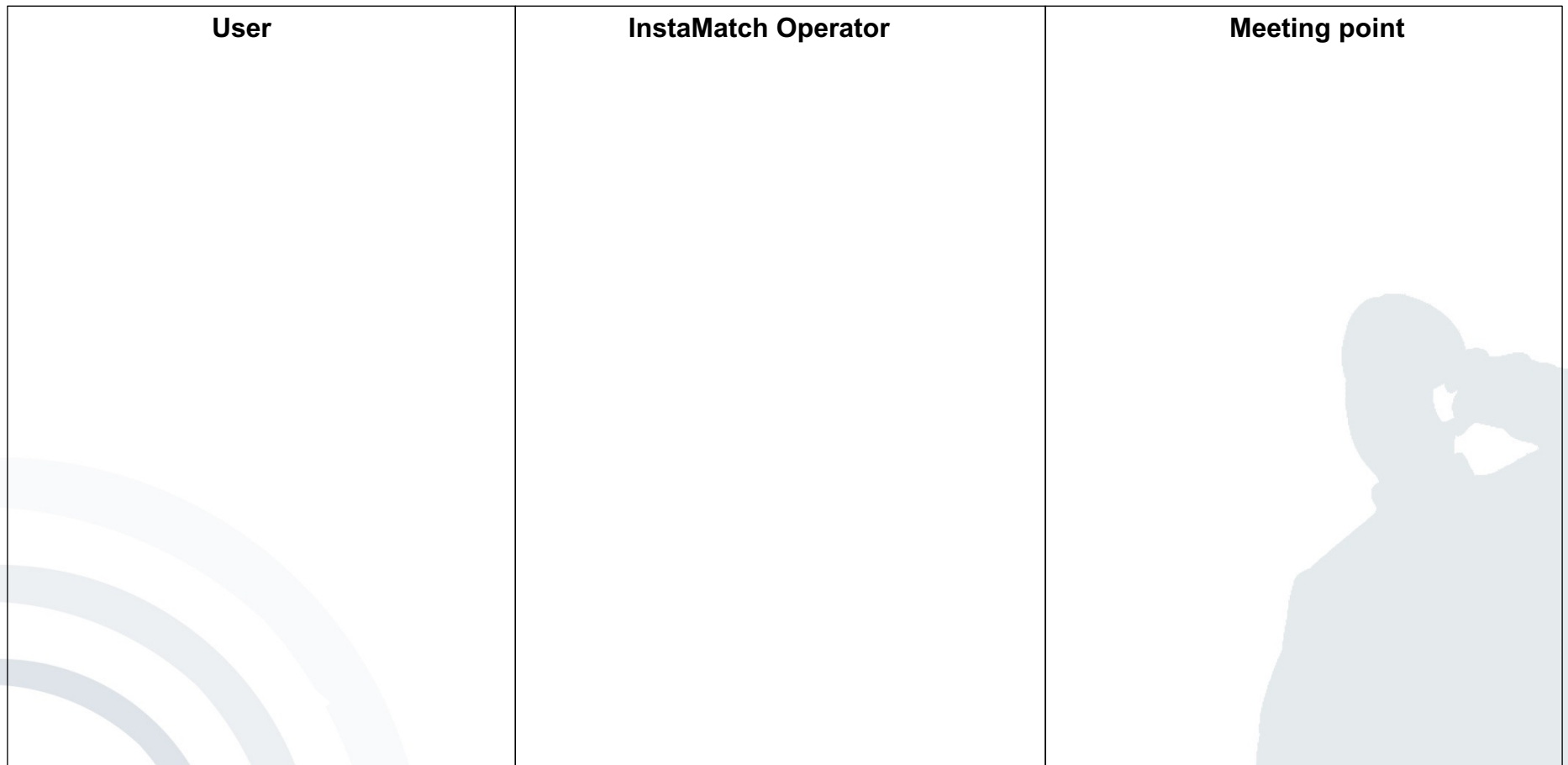
# Repetition: Activity Diagram Notation Elements

## Notation elements

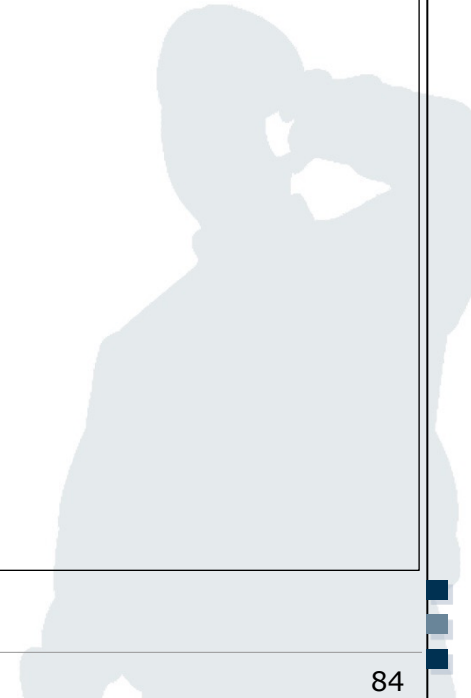
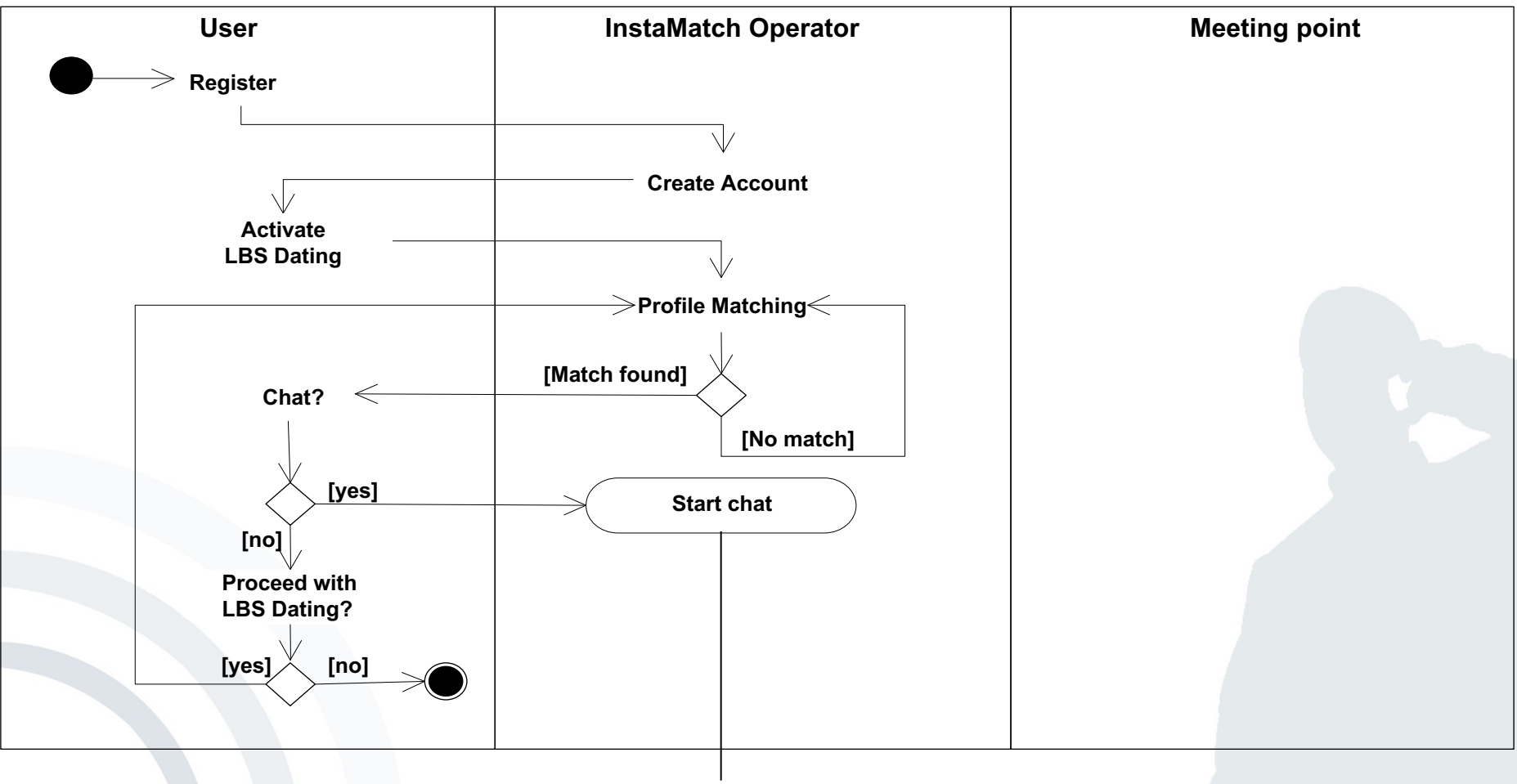
- Initial state/final state
- Activity
- Decision
- Split/join
- Responsibility
- Activity flow



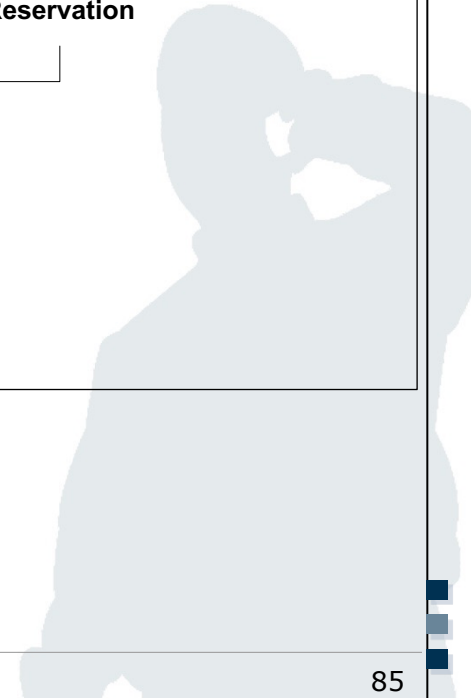
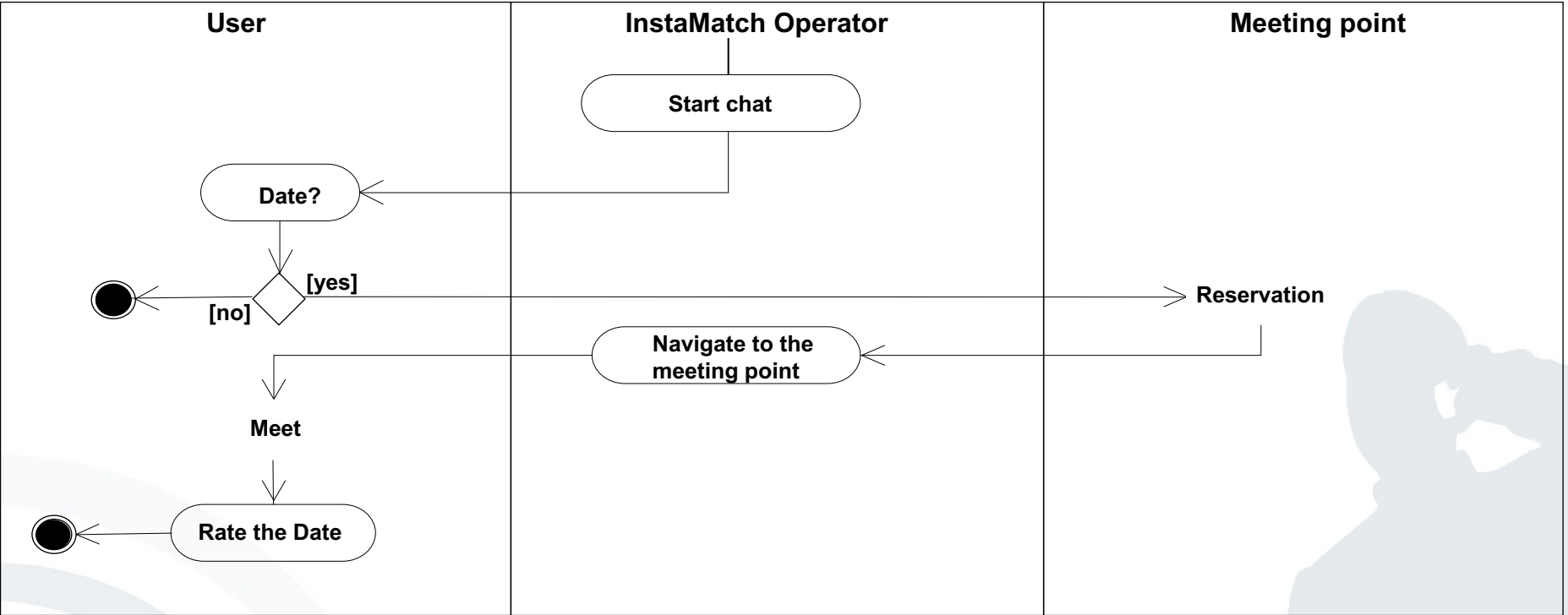
Activity diagram (1/2)



Activity diagram (1/2)



Activity diagram (2/2)



# Open Questions?