Abstract

Undergraduates need a teaching style that fits their (lack of) experience. Especially in systems engineering this is an issue, since systems engineering connects to so many different stakeholders with so many different concerns. Students need to get aware of the inherent ambiguities, uncertainties and unknowns in the systems world, in contrast to the focused world of mono-disciplinary engineering.
Well-defined problems

Well-defined methods, techniques, formalisms

Questions have the right answer

Dominant teaching style: knowledge transfer and skill training

Subject of interest is technical

<table>
<thead>
<tr>
<th>strength of bridge</th>
<th>Bode plots in control</th>
</tr>
</thead>
<tbody>
<tr>
<td>bridge will sustain upto x ton</td>
<td>control is converging and stable</td>
</tr>
<tr>
<td>no unpredictable humans</td>
<td>no inconceivable nature</td>
</tr>
</tbody>
</table>
Goals of Teaching SE at Undergraduate Level

to make students aware of
+ the impact of working in a large organization, e.g.
  processes, organizations, roles, responsibilities, economy and financials
+ the communication challenges between:
  various technical disciplines
  various less technical stakeholders
+ the ill-defined and multi-dimensional nature of system problems
  uncertainties, unknowns, ambiguities, dynamics, conflicting needs and goals
+ the impact of external conditions on the system and its design
  human behavior, natural phenomena
+ system life cycle

to provide insight in available methods, techniques, and concepts
Systems Engineering processes, methods, and techniques have a high "common sense" level. Without experience they can easily be perceived as open doors. How to get inexperienced students in a perceptive mode, such that they appreciate the subject matter?
Role Play as Teaching Paradigm

**management team**
- process
- organization
- people
- business, finance

**design team**
- customer understanding
- requirement specification
- concept selection
- partitioning, interfaces, functionality
- qualities, e.g. performance, cost, reliability
- technology choices

- teams of 3 to 5 students per team
- guided process: some theory apply in role play brief reporting and discussion
- case that relates to their knowledge
A time-box is a fixed amount of time allocated to perform one activity.

We iterate many times over different viewpoints. Every viewpoint is addressed multiple times with new insights from other viewpoints.
This didactic model is very intense. Students are exhausted after 1/2 day.

**Diagram:**

- **Classroom:**
  - Small steps on flip charts (or paper)

- **Homework:**
  - Consolidate results in PowerPoint or Visio

- **Flow:**
  - 1/2 day
  - Few days or weeks
  - 1/2 day
  - Time
Case Requirements and Example

**Case requirements**

- multi-disciplinary aspects
- original discipline of students should be clearly present
- students must have some affinity with the application
- open definition: unclear problem, large solution space

**Example: Tree Cutting Robot for mechanical engineering students**

**Mechanics is dominant**

**Vision and control require**

**Electronics and software**

**Background:**

Less young people are willing to work in the wild and mountainous areas in Norway, Canada, or USA to cut trees for wood production.

**Product:**

Robot that supports the cutting and processing of trees so that less people are needed.
Example Designs of Tree Cutting Robot
Example Stories of Apple Plucking Robot
Class Room Fills itselfs with Flip Charts
Material for Design is Based on CAFCR+

What does Customer need in Product and Why?

Customer What
Customer How
Application
Functional
Conceptual
Realization

Product How

drives, justifies, needs
enables, supports
CAFCR+ Model

Customer objectives

Application

Functional

Conceptual

Realization

Life cycle operations

operations
maintenance
upgrades

development
manufacturing
installation

Life cycle

sales, service, logistics, production, R&D

Teaching Systems Engineering to Undergraduates
14 Gerrit Muller

version: 0.1
September 6, 2020
BCAFCRplusLifeCycle
Steps for First 2 Sessions

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
Steps for Second 2 Sessions

1. make core spec
2. why are these specifications needed
3. describe usage
4. make key driver graph
5. make story
6. make use case(s)
7. analyze design impact
8. assess story based on 5 story telling criteria
9. improve story
10. improve key driver graph
11. make cost of ownership model
12. explore alternative designs
13. update specification
14. make draft management presentation
Last Session

Customer objectives

Application

Functional

Conceptual

Realization

Life cycle

1. specify life time

2. draw dev. life cycle

3. describe logistics and manufacturing

4. describe installation and acceptance

5. describe maintenance

6. update specification

7. analyze design impact
Summary of all Steps

Customer objectives

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model

Application

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model

Functional

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model

Conceptual

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model

Realization

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model

Life cycle

1. select case to work on
2. discuss possible solutions
3. discuss specification
4. make design
5. make construction decomposition
6. make functional design
7. make presentation of specification and design
8. make second and third design
9. compare three designs
10. make list of design criteria
11. make list of design choices
12. update specification
13. define performance use case
14. specify performance
15. make performance model
16. make construction decomposition
17. make presentation of specification and design
18. analyze design impact
19. make cost of ownership model
20. define performance use case
21. specify performance
22. make performance model
### Stretching from Comfort Zone into Unknowns

<table>
<thead>
<tr>
<th>Customer objectives</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
<th>Life cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. select case to work on</td>
<td>3. discuss specification</td>
<td>5. make construction design</td>
<td>7. make presentation of specification and design</td>
<td>10. make presentation of creative designs</td>
<td>13 define requirements</td>
</tr>
<tr>
<td>2. why are these specifications needed</td>
<td>4. make use case</td>
<td>6 make functional design</td>
<td>8 make second and third</td>
<td>11 update specification</td>
<td>14 make draft management presentation</td>
</tr>
<tr>
<td>3. describe usage</td>
<td>5. make story</td>
<td>6 update specification</td>
<td>9 assess story based on 5 story telling criteria</td>
<td>12 update specification</td>
<td></td>
</tr>
<tr>
<td>4. make key driver graph</td>
<td>6 improve story</td>
<td>7 explore alternative designs</td>
<td>10. make list of design criteria</td>
<td>13 define performance use case</td>
<td></td>
</tr>
<tr>
<td>5. analyze design impact</td>
<td>7 improve key driver graph</td>
<td>8 compare three designs</td>
<td>11 make list of design choices</td>
<td>14 specify performance</td>
<td></td>
</tr>
<tr>
<td>6 update story</td>
<td>8 create performance model</td>
<td>9 define performance use case</td>
<td>12 update specification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 keep iterating</td>
<td>9 specify performance</td>
<td>13 specify performance use case</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 keep iterating</td>
<td>10 make detailed model</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Teaching Systems Engineering to Undergraduates*

19 Gerrit Muller

version: 0.1

September 6, 2020

TSEUs/summaryAnnotated
Role of the Teacher

Guides the students through a journey.

Stretches students one step at a time.

Regularly forces students out of their comfort zone.

Provides feedback on their intermediate deliverables.

Helps students to reflect on their experience.

Provides theory JIT (Just In Time: appreciation and application).

Illustrates theory with examples from practice.

Keeps the pace high.

Initiates frequent breaks (this approach costs lots of mental energy).

Unfreeze students: let them sketch, stimulate creativity and imagination.
Experiences of Teaching in this Way

The sessions can be a lot of fun for students and teacher.
Some interesting concepts pop-up.
The increase of awareness can be observed.
Some nice visualizations or animations are shown.

Time-boxes can vary from 5 to 20 minutes.
Sometimes a few steps have to be skipped.

Too funny concepts or stories distract.
Teams that get stuck in unrealistic proposal.
Students that miss sessions; participation is mandatory.
Teams that stick to the initial solution
Links

Course slides:

http://www.gaudisite.nl/BachelorSDallSlides.pdf

Background CAFCR model:


Short introduction course in Systems engineering:

http://www.gaudisite.nl/ShortIntroCourseSESlides.pdf