The brontosaurus of complexity – teaching systems engineering

Dr. Jon Holt
Brass Bullet Ltd.
Overview

1. Introduction
2. Complexity
3. Systems engineering concepts
4. Case study
5. Conclusions
1. Introduction

• Systems engineering education is essential
• Lots of experience teaching systems engineering at various levels:
  – Professional/corporate/government
  – University – undergraduate/postgraduate
  – Schools – pupils and teachers
• Finally getting results?

A personal view
Problems

• Teaching a largely inexperienced audience
• Many of the reasons ‘why’ rely on experience of:
  – Working on real projects
  – Working as part of a team
  – Appreciation of key concepts
• Experience is also essential in order to teach
The challenge

• Making systems engineering real is crucial
  – Seeing the impact of failure
  – Making the connection to reality
  – Seeing the problems for themselves

• Bridging the conceptual gap in a very limited time frame
The constraints

- Limited time frame – one week
- Assume no experience
- Establish and maintain the connection to reality
- Make systems engineering memorable
- Demonstrate understanding through one-week project
The approach

• Talk about failures and disasters
• Introduce complexity
• Introduce key concepts and get them burned into student’s minds
• Investigate a famous disaster through a case study
• Present an individual famous disaster
• Investigate thoroughly through course work
2. Complexity

• One of the ‘three evils’
  – Complexity
  – Lack of understanding
  – Communication

• Chosen as the entry point into systems engineering
Complexity - example

(a)  Class 1  Class 3  Class 2
      |           |
      |           |
(b)  Class 1  Class 4  Class 2  Class 3
      |           |           |
      |           |           |
(c)  Class 1  Class 2  Class 4  Class 3  Class 5
The brontosaurus of complexity
Issues with complexity

• Relationships increase complexity
  – At all levels
  – At any point in life cycle
• The goal is the tail of the brontosaurus
  – Always start at the head
  – Must travel through belly to get there
• Complexity feeds into other evils, and vice versa
• Essential to address these issues
3. Systems engineering concepts

• There are few systems engineering concepts within the scope of the course
• Essential that they are
  – Remembered
  – Understood
• Remembered through a linking system
• Understood through case study
A linking system

1. Telephone
2. Sausage
3. Monkey
4. Button
5. Book
6. Cabbage
7. Glass
8. Mouse
9. Stomach
10. Cardboard
11. Ferry
12. Christmas
13. Athlete
14. Key
15. Wigwam
16. Baby
17. Kiwi
18. Bed
19. Paintbrush
20. Walnut
The linking system

• Remember each couplet of words as a striking visual image
  – More unusual the better
  – Try to evoke an emotion

• Helps to remember the words and identify relationships between them
Remember them – in order!
Systems engineering concepts

- Product
- Project
- Life cycle
- Life cycle model
- Stakeholder view
- Stakeholder
- Stage
- Stage instance
- Conception
- Construction
- Operations
- Development
- Transition
- Retirement
- Process content view
- Process behaviour view
- Process instance
- Artefact
- Activity
- Iteration
- Iteration
- produce/consume
- 1..*
- identifies roles involved in
- has
- is run according to
- defines behaviour for
- 1..6
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
- 1..*
Concepts and realisation

- Project
- Product
  - «Class diagram»: Life cycle
  - «Sequence diagram»: Life cycle model
- Stakeholder
  - «Class diagram»: Stakeholder view
- «Class»: Stage
  - «Life line»: Stage instance
- «Class»: Process
  - «Activity»: Artefact
  - «Activity»: Activity
  - «Activity»: Artefact
- «Activity Invocation»: Activity
- «Sequence diagram»: Iteration
- «Class diagram»: Process content view
- «Life line»: Process instance
- «Activity diagram»: Process behaviour view
- «Swimlane»: Stakeholder

Terminology:
- 'Process', an approach to doing something, that has artefacts, activities and stakeholders
- 'Artefact', the input or output to a process, that may be a document, model, code, assembly, system etc
- 'Activity', a step in a process
- 'Stakeholder', a person, place or thing that has an interest in a project and represents a distinct role
- 'Life cycle', the structural set of stages that describe a product or project from the initial idea, until its ultimate disposal
- 'Life cycle model', the behaviour of a life cycle

Modelling:
- Associations show consistency relationships
- Stereotypes (<< >>) show UML visualisation
- Elements with same names are consistent
- Dependencies show instances
4. Case study: Jurassic Park

• A case study is used as the example in the classes
  – First piece of homework is to watch the Jurassic Park movie
  – Goal to enforce familiarity with the project
• Used as a basis for analysis and discussion
  – Where did failure occur?
  – Where did problems occur?
Life cycle model - typical

- Conception
- Development
- Construction
- Transition
- Operations
- Retirement
Life cycle model – actual

- Conception
- Development
- Construction
- Transition
- Operations
- Retirement
Events

:Conception :Development :Construction :Transition :Operations :Retirement

Problem

Disaster
Process view – high level

- Process
  - Enterprise
    - Safety
    - Environmental
  - Agreement
    - Marketing
    - Payroll
- Technical
  - Research
  - Development
- Project
  - Planning
  - Monitoring
Process view – focus on technical

Requirements
  - Analysis

Design

Building
  - Coding
  - Integration

System test

Integration

Transfer
Example iteration – current (transition stage)

:Weather monitoring

:Adverse weather
Example iteration – corrected (transition stage):

- Weather monitoring
- Adverse weather
- Emergency
Individual process – existing (coding process)

Software engineer

Code software

Test

Integrate
Individual process - corrected (coding process)

Software engineer

- Code software
- [not passed]

Tester

- Test
- [not passed]
- [passed]

Integrator

- Integrate
- [not passed]
- [passed]

Reviewer

- Review
- [passed]
- [issues raised]

Release
The project

• Take a famous (real-life) disaster and apply same techniques
  – Same views generated
  – Failed processes identified and corrected
  – Missing processes identified and defined
• Work as teams for some views
• Work individually for a single iteration and process
5. Conclusions

• Problems teaching systems engineering at university level
  – Many new concepts
  – Need to bridge the gap

• Concepts are
  – Remembered through linking system
  – Understood through example and project

• Very positive feedback so far
More information

- ‘UML for systems engineering – watching the wheels’, Jon Holt, IET publishing

www.brass-bullet.co.uk

Jon.holt@brass-bullet.co.uk