

# Analysing the Assumed Benefits of Software Requirements

Richard Ellis-Braithwaite  
cordje2@lboro.ac.uk

1/17

## Introduction: The Underlying Context (1/3)

- **Who's involved?**

PhD project in collaboration with industrial partners for problem identification and solution validation:



Civil Aero (focus on bespoke software for internal use).



Technical Consulting (focus on COTS software for external use).

## Introduction: The Underlying Context (2/3)

- **What are we looking at?**

The problem was first identified while at Rolls-Royce (and then later found to exist in the MOD through working with LSC):

Software projects are often successful in that their requirements are usually met. However, the software doesn't always add the value that was expected/hoped for.

## Introduction: The Underlying Context (3/3)

- **Which leads us to ask:**

“How can we model the value of software before it exists..  
so that its value can be aligned with organisational strategy..  
and so that stakeholder expectations can be sanitised?”

# Introduction: The Research Questions

- RQ1. What evidence exists to show that implemented requirements (features/qualities) are not always beneficial?
- RQ2. What is an appropriate approach for modelling the assumed benefits of software requirements?
- RQ3. What aspects of the resulting benefit model are important for analysing the strategic alignment of software requirements?
- RQ4. What are the quality characteristics of such models, and what challenges preclude them?
- RQ5. How can a supporting tool address the challenges elicited from RQ4?

5

## Motivation from the Literature

**Summary:** lots of waste occurs in software development/acquisition, and it's not just the customer who pays for it! *Satisfied requirement != +value.*

<b>Key Message</b>	64% of delivered software functionality is never/rarely used (45%, 19% respectively).	73% of COTS software is never used, which leads to "bloated software" that is hard to maintain and to use.	There is little to no correlation between a company's level of IT investment and its profitability.
<b>Source</b>	Chaos Report v3 Analysis, Scott W. Ambler, 2006	"Bloat": the objective and subject dimensions, J. McGrenere, CHI 2000	Does IT Matter? Information Technology and the Corrosion of Competitive Advantage, HBR Article, N.G. Carr, 2004

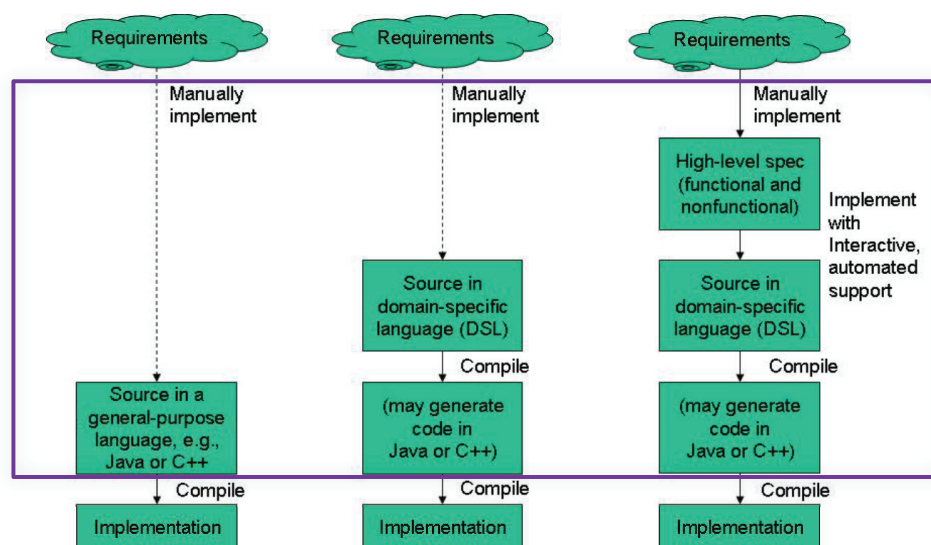
6

# Motivation from the Industrial Partners

- Managers need to be shown **how** software systems **align with the objectives** of the business – “singing from the same hymn sheet”.
- Questions such as “**Why** do we need this function, and **why** should its output be this precise?” are sometimes hard to find the answer to.
- Business stakeholders **don’t understand the application domain**, but decisions made there impact the satisfaction of their objectives. “I thought the software would make analysis faster AND more accurate!”.
- Stakeholders believe that **their requirements are the most important** because they only know their domain (they are specialists) – priority should be based on business needs, not the interests of engineers.

7

## Tracing Below a Requirement is Important

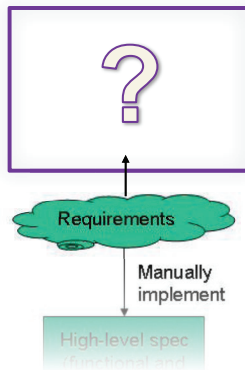


Because the utility of the solutions depends on the requirements.

Levels of Abstraction in Model Driven Architecture

8

# The Same Applies for Tracing Up



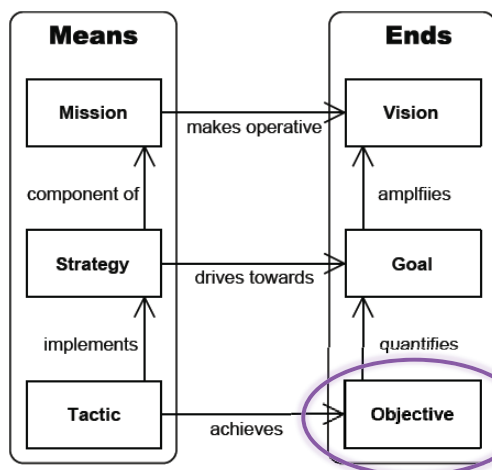
Because the utility of requirements depends on what's above them (i.e., needs / desires / problems).



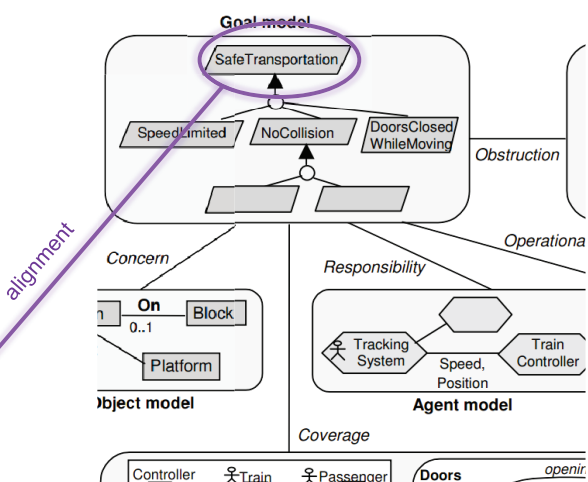
9

# Alignment: System & Business Goals

Goals/objectives are just problems phrased positively.  
So, how will the system's goals contribute to the organisation's problems?



OMG's Business Motivation Model (bounded by the vision of the overall organisation).



System Goal Model (bounded by the ability of the system's agents to influence the goals).

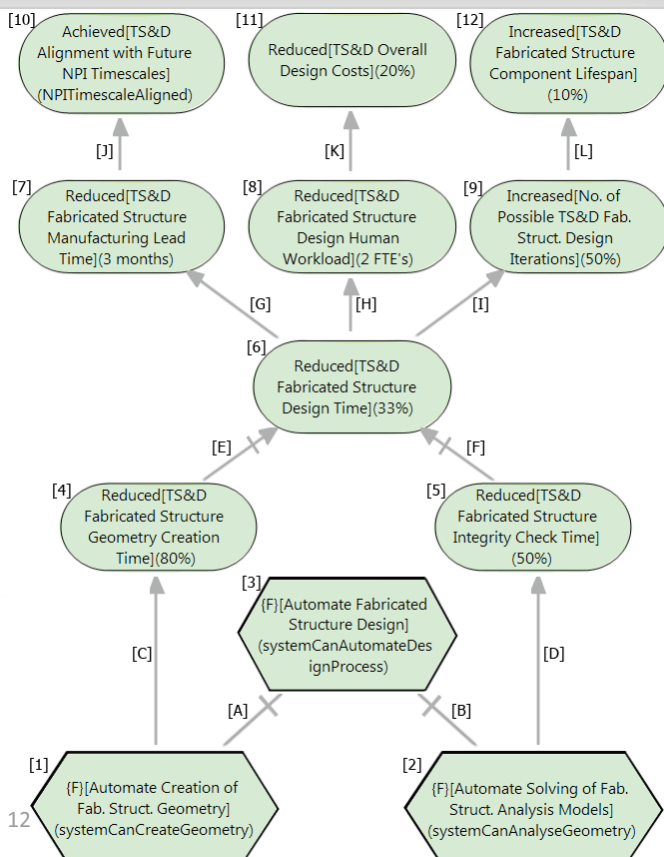
10

# Brief Critique of Current State of the Art

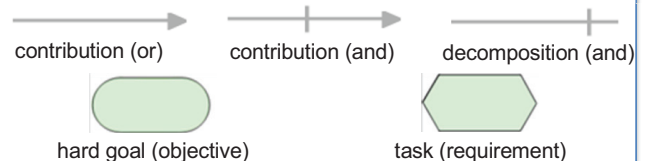
- B-SCP Requirement Strategic Alignment Framework  
Does not use contribution weights of any kind to represent the extent of the alignment, e.g., requirementX supports objectiveC - but how well?
- Goal Oriented Requirements Engineering (GORE), e.g., i\*, KAOS  
Goal-goal contribution is not considered in terms of the effects some contribution has all the way up the goal chain, e.g., some satisfaction of requirementX to extent y in terms of objectiveC, objectiveB & objective A.
- House of Quality Diagram (QFD)  
Does not use application domain metrics to explain the contribution made – how can we verify that requirementX supported objectiveC by “6” (on a scale of 1-9). Does not abstract goals (e.g., why is objectiveC important?).
- None consider confidence, despite the uncertainty involved in predicting that requirementX will support objectiveC to some extent.

11

## Our Method: An Example Alignment Diagram



### Goal Graph Diagram Key



### Objective Definition Template (GQM) - Item #7

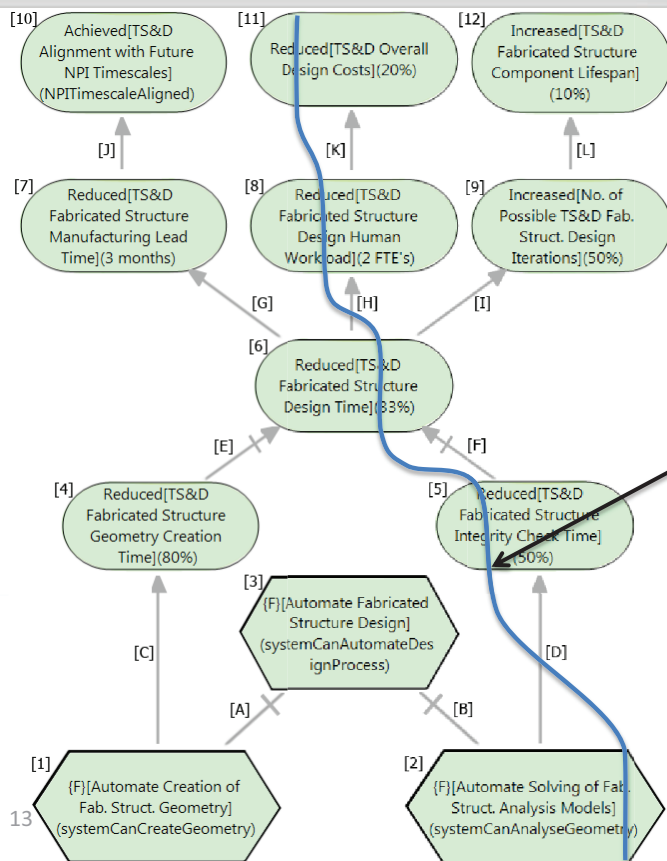
Activity	Reduced
Object	TS&D Fabricated Structures
Focus	Average Manufacturing Lead Time
Magnitude	3 months
Scale	Average time in months required to have FS parts manufactured from the inception of a new engine
Timeframe	1 year after system deployment
Scope	Transmissions Structures & Drives (TS&D) SCU
Author	John Smith (Component Engineer, TS&D)

### Contribution Specification Examples

Link	[Contribution] [Activity] [Scale]	Confidence
C (1→4)	[80%] [Reduction] in [Geometry Creation Time]	1
D (2→5)	[50%] [Reduction] in [Integrity Check Time]	0.75
E (3→6)	[33%] [Reduction] in [Design Time]	0.5

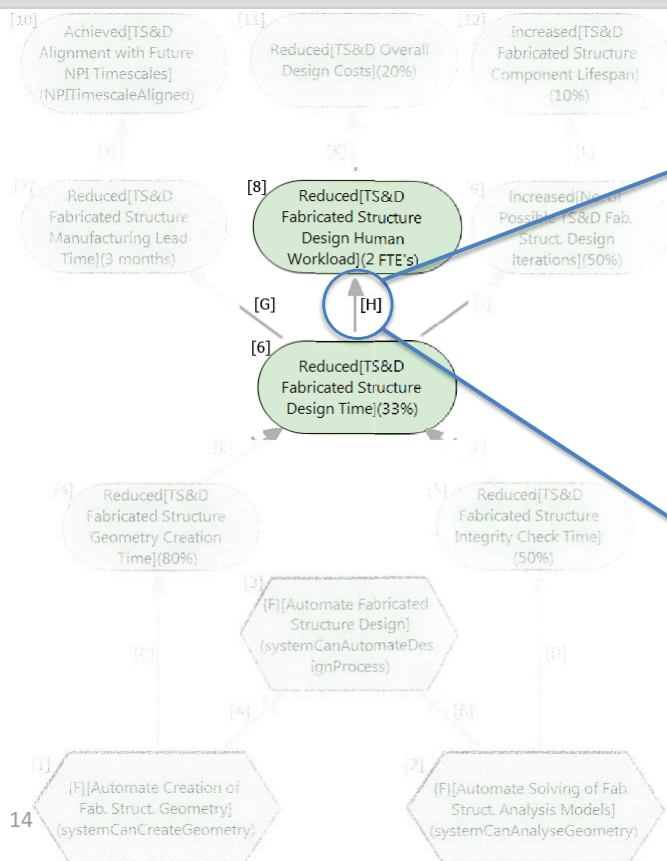


# Dynamics of Contribution Links

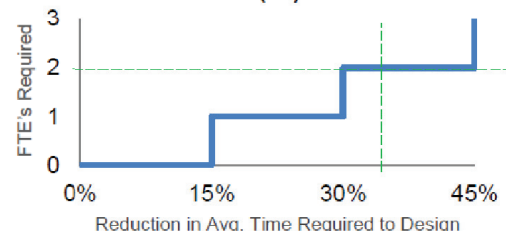


If this point (or any) changes, the entire chain of cause/effect does too. (Where the line's point on the x-axis of a goal represents the extent of the goal's satisfaction.) Thus, the previous table showing link contributions is only valid if each of the goals is satisfied to the specified target level (unlikely).

# Better Describing the Contribution Links

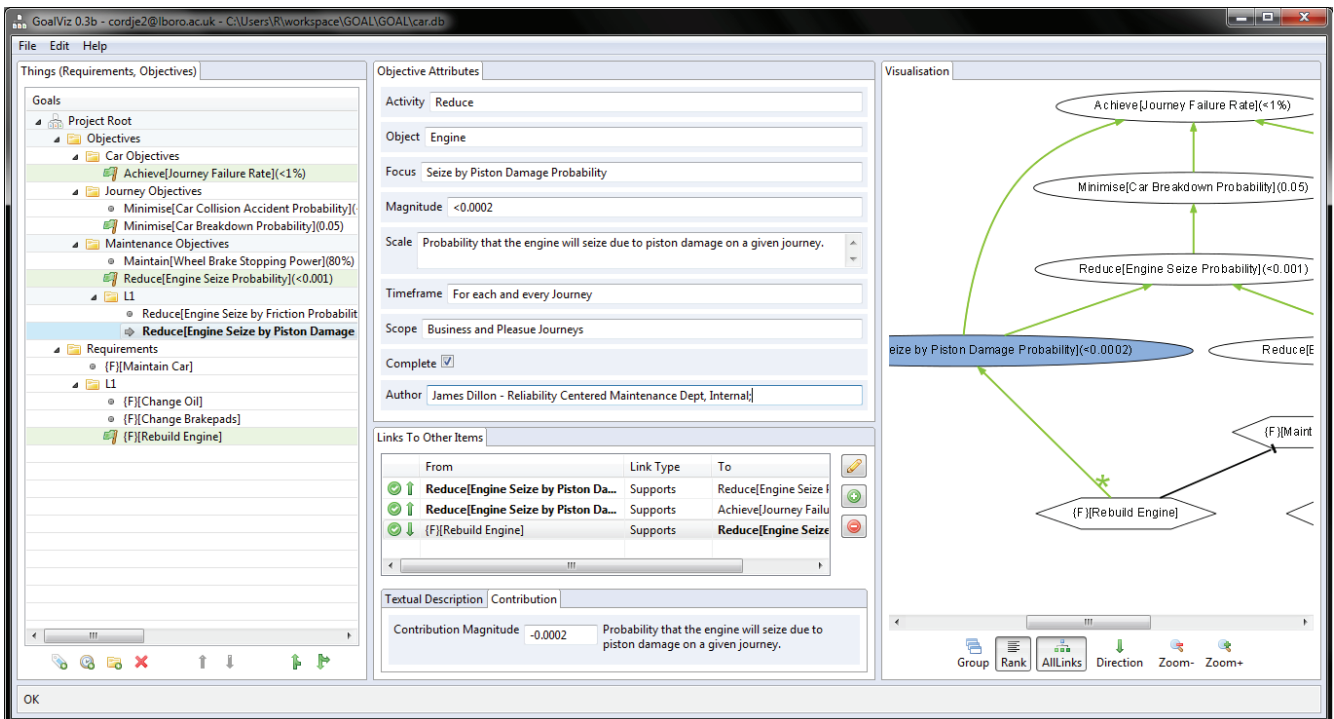


**Link (H):** The effect on the number of "Full Time Employees (FTE's) required for TS&D FS Design" when the "Avg. Time Required to Design Fabricated Structures (FS)" is reduced.



"What if?" questions..

# GoalViz Tool Screenshot (0.3b)



15

## Evaluation & Future Work

The construction of the evaluation framework to judge the usability and utility of the approach is in progress.

- It is a challenge to elicit observable phenomena that represent the benefits of applying the approach; we don't have time to wait for software to be developed & deployed.

Future work is to improve the accuracy of goal-goal contributions by:

- using stakeholder networks (as in StakeSource) and "wisdom of the crowd" theory to capture multiple sets of contribution forecasts;
- using similarity analysis on previous projects to find similar data (evidence) to base estimates upon.

16



# Closing Remarks

- Challenges/difficulties:
  - There's a good reason that contribution scores like {Low, Medium, High} are used! (normalised & doesn't require data).