

Systems engineering: challenging complexity



Systems engineering: Challenging complexity



OpenLearn

Free learning from
The Open University

This item contains selected online content. It is for use alongside, not as a replacement for the module website, which is the primary study format and contains activities and resources that cannot be replicated in the printed versions.

About this free course

This free course provides a sample of level 3 study in Computing & IT

<http://www.open.ac.uk/courses/find/computing-and-it>

This version of the content may include video, images and interactive content that may not be optimised for your device.

You can experience this free course as it was originally designed on OpenLearn, the home of free learning from The Open University:

<http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/systems-computer/systems-engineering-challenging-complexity/content-section-0>.

There you'll also be able to track your progress via your activity record, which you can use to demonstrate your learning.

The Open University Walton Hall, Milton Keynes MK7 6AA

Copyright © 2023 The Open University

Intellectual property

Unless otherwise stated, this resource is released under the terms of the Creative Commons Licence v4.0 http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_GB. Within that The Open University interprets this licence in the following way:

www.open.edu/openlearn/about-openlearn/frequently-asked-questions-on-openlearn. Copyright and rights falling outside the terms of the Creative Commons Licence are retained or controlled by The Open University. Please read the full text before using any of the content.

We believe the primary barrier to accessing high-quality educational experiences is cost, which is why we aim to publish as much free content as possible under an open licence. If it proves difficult to release content under our preferred Creative Commons licence (e.g. because we can't afford or gain the clearances or find suitable alternatives), we will still release the materials for free under a personal end-user licence.

This is because the learning experience will always be the same high quality offering and that should always be seen as positive – even if at times the licensing is different to Creative Commons.

When using the content you must attribute us (The Open University) (the OU) and any identified author in accordance with the terms of the Creative Commons Licence.

The Acknowledgements section is used to list, amongst other things, third party (Proprietary), licensed content which is not subject to Creative Commons licensing. Proprietary content must be used (retained) intact and in context to the content at all times.

The Acknowledgements section is also used to bring to your attention any other Special Restrictions which may apply to the content. For example there may be times when the Creative Commons Non-Commercial Sharealike licence does not apply to any of the content even if owned by us (The Open University). In these instances, unless stated otherwise, the content may be used for personal and non-commercial use.

We have also identified as Proprietary other material included in the content which is not subject to Creative Commons Licence. These are OU logos, trading names and may extend to certain photographic and video images and sound recordings and any other material as may be brought to your attention.

Unauthorised use of any of the content may constitute a breach of the terms and conditions and/or intellectual property laws.

We reserve the right to alter, amend or bring to an end any terms and conditions provided here without notice.

All rights falling outside the terms of the Creative Commons licence are retained or controlled by The Open University.

Head of Intellectual Property, The Open University

The Open University

Martins the Printers, Berwick-upon-Tweed

Contents

Introduction	6
Learning Outcomes	7
1 Why is systems engineering important?	8
1.1 Introduction: what is the problem?	8
1.2 The Phoenix project	11
1.3 Example 1 The Workcenter that didn't	13
1.4 Example 2: The Bridge of Sighs (and Wobbles)	18
1.5 Increasing complication, complexity and risk: the underlying relationship	26
1.6 Increasing complication, complexity and risk: mystery and mechanics	30
1.7 Increasing complication, complexity and risk: a spectrum of systems intractability	33
1.8 Increasing complication, complexity and risk: are systems becoming more complex?	38
1.9 Increasing complication, complexity and risk: summary	42
2 What is engineering?	43
2.1 The development of engineering	43
2.2 A modern view	45
2.3 Summary and conclusions	49
3 What is systems?	50
3.1 Introduction	50
3.2 Systems concepts: system	53
3.3 System concepts: holism	58
3.4 Systems concepts: structure	62
3.5 Systems concepts: dynamic behaviour: input-transformation-output	64
3.6 Systems concepts: dynamic behaviour: control	67
3.7 Systems methodologies for managing change	70
3.8 Systems methodologies for managing change: hard systems approach	70
3.9 Systems methodologies for managing change: soft systems approach	73
3.10 Systems techniques	78
3.11 Summary	78
4 What is systems engineering? The career of a concept	80
4.1 Beginnings	80
4.2 The use of systems analysis in public policy	80
4.3 The use of systems engineering in organisations	83

4.4 The use of systems engineering in organisations: different organisational arrangements	84
4.5 Methodologies associated with information technology	87
4.6 Systems engineering: the recent development of a discipline	92
4.7 Summary	99
5 The original course team's approach to systems engineering	100
5.1 Introduction: the general framework	100
5.2 The aims and principles of system engineering	101
5.3 The systems engineering methodology used in the course	105
Conclusion	108
Keep on learning	109
References	109
Acknowledgements	112

Introduction

The aim of this course is to answer five questions:

- Why is systems engineering important?
- What is modern engineering?
- What is systems?
- What is systems engineering?
- What approach to systems engineering does the course adopt?

This OpenLearn course provides a sample of level 3 study in [Computing & IT](#)

Learning Outcomes

After studying this course, you should be able to:

- evaluate a specific example or case of a product development process in terms of the 'waterfall' life cycle model of software development
- classify new product developments as: fault correction, enhancements, new but similar products, radically different, revolutionary or iconoclastic products
- analyse the causes of a systems failure
- identify and evaluate the importance of the relationships of the factors leading up to system complication and complexity
- answer the question 'why is systems engineering important?'

1 Why is systems engineering important?

1.1 Introduction: what is the problem?

In late June and early July 2005 a row erupted concerning the operation of a major flagship of government social policy, the tax credit system. Introduced in 2003, it was designed to help those on low incomes and whose social circumstances prevented them from working full-time (Citizens Advice Bureau, 2005). The article reprinted in Box 1 indicates the extent of the political unrest with a system that left families relying on food parcels, and that has been variously described as being 'in chaos' and 'shambolic'. Such problems have become a familiar story and the steady stream of failures seems destined to continue into the future.

During the same period a committee of members of parliament issued a warning that of 254 government-funded computer projects currently under development, 70 had been given a 'red light', meaning that they will fail to deliver the promised benefits unless immediate action is taken. Of these 70 projects, 8 had been given 'double-red warnings' (*Guardian*, 5 July 2005).

There is, however, nothing inevitable about such failures, as the example of London's congestion charging scheme demonstrates. In February 2003 those entering a 22 km² zone in central London in a private car between 7 a.m. and 6 p.m. were liable to pay a congestion charge of £5. The scheme is enforced by a network of over 700 cameras at 203 sites located at all entry and exit points to the zone, plus some additional mobile patrol and other units. The analogue data from these cameras is streamed back to a central hub and is put into an automatic number plate recognition system which then checks details against prepayment, the Driver and Vehicle Licensing Centre database and, when necessary, sends out penalty notices and manages revenue collection. The system is designed to cope with up to '250,000 vehicles [making] 450,000 movements into the charging zone during the period 7 a.m. to 6.30 p.m. with 40,000 vehicles an hour driving into the congestion charging zone during the morning peak (7 a.m.–10 a.m.)' (Transport for London, 2005). This complex system combining a number of different technologies was developed and became operational within 18 months.

Box 1 Chaotic scheme that left families relying on food parcels

Demands mount for overhaul of aid that led to 'debt and despair'

Gordon Brown, the chancellor, faced fresh calls last night for a radical overhaul of the tax credit system after the Treasury published figures showing that administration costs have risen tenfold in four years to £400m.

The figures also revealed that 65.5 million award notices have been issued in just over two years – more than 10 on average for each of the 6 million households that receive tax credits.

Critics said the system was in chaos and the figures undermined the Treasury's insistence that a series of improvements were beginning to take effect.

The Liberal Democrat spokesman David Laws, who obtained the figures in answers to parliamentary questions, said low-income families were suffering severe hardship and ministers appeared to be immune to their appeals for help. He said fundamental

flaws in the system meant hundreds of thousands of low-income families would still face 'debt and despair'.

To claim child tax credit and working tax credit, families must predict their income for the next financial year and tell the Inland Revenue about any changes in circumstances. Payments can reach £5,000 to £6,000 but many have been forced to pay back some or all of their tax credits.

Backbench MPs, who wrote more than 5,000 tax credit complaint letters in the first five months of this year, say many families persuaded to leave benefits and return to work by promises of tax credits take on extra commitments, including childcare costs, only to find that computer or human errors at the Inland Revenue's tax credit centre in Preston result in an overpayment and a demand for the money to be returned.

Frank Field, a former social security minister, tabled a motion in parliament yesterday calling for an independent appeals tribunal to hear cases before the Revenue claws back overpayments.

Ministers revealed that 217,000 households asked for overpayments to be written off in 2004/05 and 54,000 have already made write-off requests in the first two months of the 2005/06 financial year.

Dawn Primarolo, the paymaster general, who provided the answers, admitted last month that the system needed to become more sensitive to the needs of low-income families. David Varney, chairman of Revenue and Customs, admitted that the tax credit structure may need to be overhauled if problems persist.

He told the *Financial Times* that the policy of annualised tax credits should be reconsidered if it created the same problems year after year. His comments followed criticism from several quarters last month, including the parliamentary ombudsman and Citizens Advice.

Citizens Advice said poor administration and the recovery of overpaid tax credits had led to some families living off only £56 a week plus child benefit.

In the most extreme cases, families had been threatened with repossession or eviction, and bureau staff had to arrange Salvation Army food parcels for some who were unable to buy groceries.

The parliamentary ombudsman suggested that Revenue and Customs should write off benefits mistakenly paid to families rather than try to claw them back. In 2003–04 about 1.9 million families were overpaid almost £2bn, either because they did not report rises in income or because of administrative errors. Many were forced to repay the money, causing hardship because most had spent it.

Mr Laws said yesterday's figures also showed the £1.9bn official cost for overpayments was understated by £800m because of sums not recovered owing to a £2,500 'disregard' when income rises. So far, only £37m of overpayments have been written off.

The data also shows that 21,600 compensation payments have been made since April 2004 for bad service.

Ms Primarolo said administration costs had remained at 2.5% to 3% since 1999.

But Mr Laws called for urgent reform. 'The tax credits system is an increasingly expensive mess. Gordon Brown's flagship tax credits scheme seems to have turned into a bureaucratic nightmare,' he said.

'What we now have is a system which is complex and bureaucratic to administer and in which overpayments are endemic – driving people on low incomes into debt and despair.

'The chancellor should consider bringing back fixed half-yearly awards, as were used with family credit and working families' tax credit. Until major reform takes place, the current chaos will continue.'

The Treasury said most problems related to computer and administrative errors in the first few months of the system.

A spokeswoman said: 'If you are introducing an entirely new tax credits scheme and going from zero to 80% take-up then of course the absolute costs of administering the scheme are going to rise.

'The key measure is whether the administration costs as a proportion of the total payments have risen or fallen, and in the case of tax credits, they have actually fallen from 3.3% in 1999 to 3% in 2004.'

Comedy of errors

Families who have appealed against demands by the Inland Revenue for the return of tax credits say an automated system cannot cope with the complexities of real life.

Bernadette Reddy-Oaten, a divorced mother of two who lives in Torquay, received an award for child tax credit which the Revenue later decided was too high. She had already spent the money on child care when the Revenue realised its mistake. It demanded she return the cash and even threatened to send compliance officers to 'interview' her and check on her finances.

Last year, after the Revenue stopped her tax credits, Ms Reddy-Oaten had to survive on 'hardship' payments. These payments were then deemed excessive and she faced demands that these also be returned, even though one was as compensation.

'I have spent the last two years trying to settle the situation after nearly losing my home in 2004 when they stopped my payments. This is despite letters acknowledging their fault,' she said.

'It's shocking when you ring the main tax credit help desk because the staff know very little about the system. It's no wonder the system costs so much to run when I receive two or three award notices a month, usually on the same day, always with different figures, and I have had to ring them weekly to resolve the situation. This happens each year. They end up admitting they owe money and pay it back, only for it to go wrong again.

'It's an understatement to say it has been a comedy of errors. And I'm a social worker. I know many of my clients have had terrible trouble, standing in phone boxes for hours trying to get through to the help desk.'

Last week the Revenue apologised for mistakes and cancelled the latest demands. She says the Revenue still owes her money but staff tell her they cannot override the computers to correct many of the errors she has suffered.

Critics inside the Revenue say the computer and call-centre systems were basic and unable to cope with the level of inquiries from low-income families, many facing marital strife and redundancy.

Charities suggest that since April tax officials have been allowed by the Treasury to write off overpayments dating back more than two years in an effort to clear a significant number of cases.

Ms Reddy-Oaten said: 'As a social worker, I know the system well and am articulate and confident enough to repeatedly challenge the Inland Revenue; I know too many families who would find it impossible to get what they are owed.'

Source: Phillip Inman, the *Guardian*, Wednesday 6 July 2005

1.2 The Phoenix project

It is all too easy to dismiss problems like that being experienced with the tax credit system as being inherent in the design and implementation of computer-based systems. But they are not restricted to computer systems, as the example of the Phoenix project in Box 2 shows.

Box 2 Fly-away drones put robot air force off course



The dismal failure of a British built drone during the war in Iraq has led the Ministry of Defence to reconsider its plans for a futuristic fleet of unmanned aircraft. Britain lost 23 of its Phoenix surveillance planes, which each cost about £1.5m.

Military tacticians believe drones will play an increasingly important role in 21st century conflicts. In Afghanistan drones were used to find and attack Al-Qaeda targets. An armed US Predator was also used in Yemen to assassinate a senior Al-Qaeda fugitive.

But there is growing controversy over Britain's preferred choices for an £800m drone air force. The Phoenix has such an abysmal record that the army, which uses it for artillery spotting, has called it 'the Bugger Off' because of its tendency never to come back.

Since 1996 198 drones have been delivered to the armed forces, but many have been cannibalised for spare parts, some have been mothballed and a large proportion of the working fleet was lost because the planes crashed, were shot down or just went missing in Iraq.

Britain is not buying any more Phoenixes, but despite the proven success of the American-built Predator the defence ministry decided earlier this year that it would buy either an Israeli-designed drone or a US unmanned reconnaissance helicopter. Both would need substantial modification if they were to carry weapons.

The Phoenix – named after a mythical bird that rises from the ashes, rather than one that crashes and burns – caused problems from the start. Development began in 1982 and it took 16 years before the plane entered service.

It has to land on its back because sensitive surveillance equipment was mounted on its underside. During trials it landed too heavily, damaging the fuselage.

Exasperated engineers fitted an airbag, similar to those found in cars, to cushion the impact. The development programme alone cost £250m.

Another problem is that the Phoenix is launched by catapult from a vehicle that is so heavy it cannot usually be flown into a battle theatre and has to be sent out from Britain by sea.

By contrast, American forces lost only four Predators during the war in Iraq. Two were deliberately wasted by being flown over Baghdad to draw fire until they ran out of fuel. One was shot down over the Tigris and Iraqi television showed pictures of soldiers and militia searching the riverbank for the pilot of what they thought was a manned warplane.

The Phoenix fiasco has led some senior military personnel to urge Geoff Hoon, the defence secretary, to think again and buy an off-the-shelf design with a proven record.

Air Commodore Ron Cook, who was last month appointed director of equipment capability for intelligence, surveillance, target acquisition and reconnaissance, has told colleagues that he is unhappy with the present choices and would like them to be reconsidered.

The two shortlisted designs are the Fire Scout helicopter and the Silver Arrow. The Fire Scout was built for the US Navy and Marine Corps, but only in small numbers because of concerns about its vulnerability. It is slower than fixed-wing designs and can be easier to shoot down.

The Silver Arrow is built by Elbit Systems of Israel for the Israeli military. It is being offered to Britain by the French arms contractor Thales.

Critics say the Ministry of Defence is accident-prone when it buys equipment. Hoon has inherited and presided over what insiders describe as a 'zoo' of turkeys, white elephants and dodos. Other fiascos include the new Apache attack helicopter. Some of the 67 aircraft ordered, which cost £27m each, are sitting in hangars because of a software problem on training simulators which means there are too few pilots trained to fly them.

Even the SA80, the army's standard issue rifle, caused years of problems. It performed well in Iraq but when it first entered service in 1986 soldiers said it often jammed and it had to be modified several times at a cost of £90m.

Source: *Sunday Times*, 22 June 2003

Software and software-focused systems of the sort highlighted in Box 1 seem to create severe difficulties when it comes to delivering what is required, when it is required and at the estimated cost. As Brooks (1987) put it:

Of all the monsters that fill the nightmares of our folklore, none terrify more than werewolves, because they transform unexpectedly from the familiar into horrors. For these, one seeks bullets of silver that can magically lay them to rest.

The familiar software project, at least as seen by the non-technical manager, has something of this character; it is usually innocent and straightforward, but is capable of becoming a monster of missed schedules, blown budgets, and flawed products. So we hear desperate cries for a silver bullet – something to make software costs drop as rapidly as computer hardware costs do.

But, as we look to the horizon of a decade hence, we see no silver bullet. There is no single development, in either technology or in management technique, that by itself promises even one order-of-magnitude improvement in productivity, in reliability, in simplicity.

It is nearly twenty years since these gloomy predictions were made, but the example reported in Box 1 suggests that we have still not found the silver bullet with which to lay the werewolves of software-focused systems developments to rest. Equally, the example in Box 2 of the Phoenix unmanned surveillance aircraft shows that failure is not confined to software projects.

The problems that lead to the partial or complete failure of a complex systems *project* can be grouped into three categories:

- a misunderstanding or uncertainty about the 'wants' being addressed (see Example 1 below)
- an inability to design a system that will meet a requirements set (Example 1)
- an incomplete or poorly executed implementation (Example 2).

1.3 Example 1 The Workcenter that didn't

Autodesk Inc. is the world's largest supplier of design and engineering software. It currently markets over thirty products but is most famous for its AutoCAD® two- and three-dimensional design and drafting software. The company is the market leader in this type of application, with over 4 million customers worldwide.

The Autodesk story began in 1982 with a group of programmers, centred on San Francisco, writing code for design software in their spare time. The group demonstrated a cobbled-together revision of what was to become AutoCAD® at an exhibition in March of the same year. The product was finally launched formally in December and was an instant success. By 1987, 100,000 copies of the application had been sold and, by 1999, 2 million. The reasons for AutoCAD's success are:

- It addressed a known, well-understood and fairly standard need – two-dimensional and later three-dimensional drafting. In this respect it was similar to word processing.
- The application is widely needed. Design and drafting are undertaken in businesses large and small throughout the world.
- It made use of the growing power and capability of personal computers. Previously, CAD packages had required a mainframe computer and the few workstations connected to them were situated in special 'suites'.
- It was (relatively) easy to use. The extensive training required to use mainframe-based CAD packages was not needed.
- It was (therefore) affordable and was sold as a standard off-the-shelf package in software retailers.

The company employs over three thousand people worldwide and generated revenues of \$820 million in the financial year 1999/2000 (Autodesk, 2000). This example concerns four new or enhanced products that were developed during the period 1992–96.

For a while Autodesk, though anxious to develop new products, had a development approach that was described as 'chaotic'. There was no formal development model or procedure, and documentation was produced as an afterthought. Recognising the problems that arose as a result of this informal way of developing new products, the company developed a product definition process (PDP). This had the stages shown in [Figure 1](#).

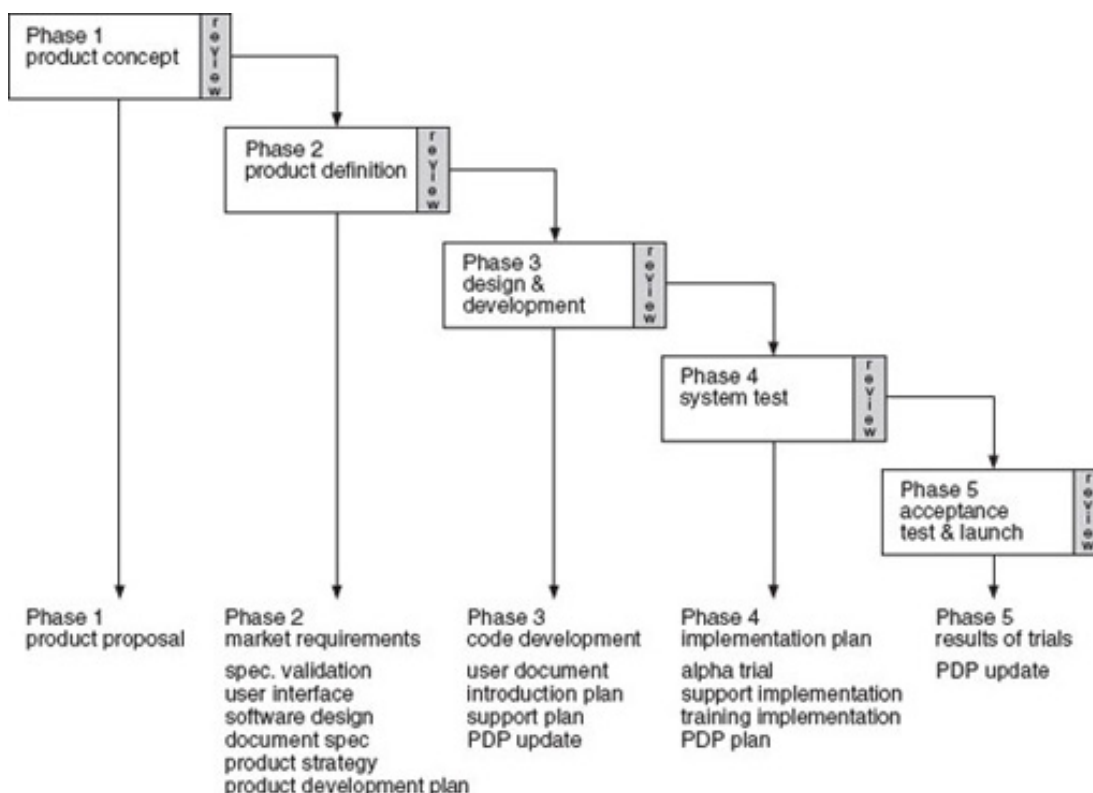


Figure 1 Autodesk's product definition process

Autodesk's PDP can be characterised as a simplified version of the standard 'waterfall' life cycle model of software development shown in [Figure 2](#).

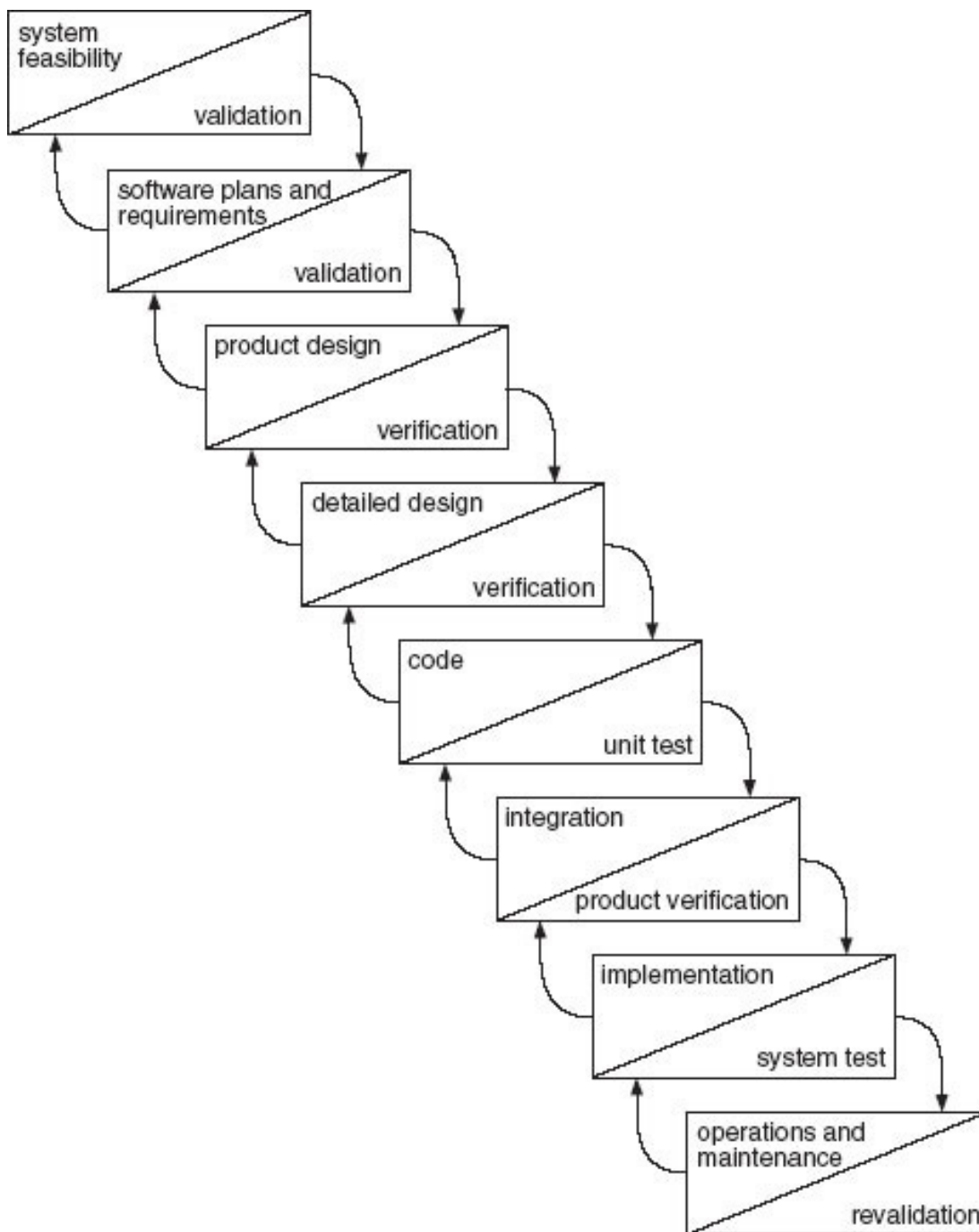


Figure 2 The waterfall life cycle model of software development. Source: Eisner (1988)

SAQ 1

What do you consider to be the major differences between Autodesk's PDP and the waterfall model depicted in [Figure 2](#)?

Answer

Although Autodesk's approach has fewer stages than Eisner's version of the waterfall model, the major difference seems to be the absence of any feedback

mechanism between the stages of the Autodesk model. Another difference is the review process at the end of each of Autodesk's stages. However, in reality, the role of the review processes was simply to ascertain the completion of each set of documents and did not consider whether their content confirmed that the work undertaken during the stage had been undertaken competently (Abram, 2001). In this respect a parallel can be drawn with many implementations of the quality standard ISO 9000, which seemed more concerned with ensuring that processes had been accurately documented rather than whether they had been competently carried out.

Autodesk used the PDP to guide the development of four new products during the period 1992–96:

- AutoCAD Release 13: an update of the original Autodesk package that provides designers and engineers with a personal-computer-based 2D and 3D design and drafting environment and toolset.
- AutoCAD LT: a version of AutoCAD with reduced features, supporting 2D drafting and design only.
- Mechanical Desktop: an add-on package for users of AutoCAD that provides additional features for mechanical designers and engineers.
- Workcenter: a document/data management package implemented on a client-server configuration.

Of these products, three were successful and remain in the Autodesk product portfolio but the fourth, Workcenter, failed to sell in sufficient numbers and was dropped. There were three main reasons for the failure of Workcenter.

First, AutoCAD Release 13, AutoCAD LT and Mechanical Desktop were all closely related to Autodesk's existing product. None of them was a significant innovation. Certainly each had new features, but their basis, application area, market and user bases were already well known to Autodesk. This was not the case with Workcenter. AutoCAD and its derivatives are used by designers and engineers essentially on a standalone basis. Workcenter was intended to integrate the work and management of the different people and departments involved in a project. Its user base was less clear.

Second, as market leader in design and engineering software, Autodesk's file format and interfaces had become a de facto standard. However, document management software has to be capable of accepting many different types of file, produced by a variety of applications. This complicated the technical design of the package enormously.

Third, as suggested earlier, drafting and design processes, at a detailed level at least, are well denned. Engineering drawings are pretty much standard and the way in which they are produced is also standard. They are the graphic equivalent of word processing. This is not the case with document management, where each organisation has its own highly individual way of doing things. The implication of this difference is that whereas a design and drafting package can be used more or less as it stands from the box, the way in which Workcenter was implemented needed to be thought through in a great deal of detail and had to take into account the way that the whole organisation worked. Each implementation had a high degree of individuality. Autodesk's distribution and sales channels were not used to providing the degree of support needed to implement Workcenter, nor was it clear how the extra work was to be paid for.

Sticking with the existing situation, the status quo, carries least difficulty and risk (see [Figure 3](#)), though it is often, correctly, argued in relation to business development that to

'do nothing' is not an option. There are, after all, plenty of examples of enterprises that did nothing only to be driven out of business by their more active competitors. However, at the level of the individual activity, carrying on as normal is least risky.

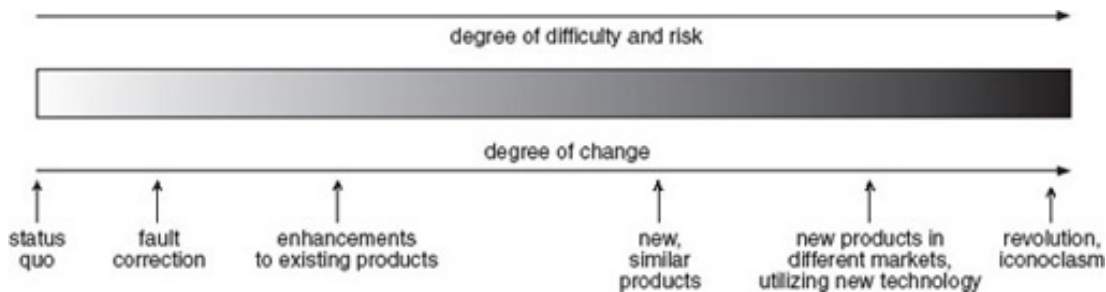


Figure 3 Degrees of difficulty, risk and change in new product development

At the second level of difficulty and risk is fault correction. Any interference in the operation of a system carries some risk, and the phenomenon of engineer-induced faults is well known. Changing an existing product, even where there is presumed to be an enhancement, can be dangerous as the alterations made previously by Coca-Cola to its formulation have shown. However, Coca-Cola have successfully, and paradoxically, introduced several similar products such as diet, cherry and vanilla versions of its standard soft drink. Generally, though, introducing a new product carries greater risk and difficulty and this is exacerbated if it involves new markets or technologies. Finally, the outcomes of attempts to produce revolutionary products that overturn existing thinking and mores is extremely risky and difficult. The use of systems engineering and an holistic approach can help reduce risk and difficulty in all these types of change and becomes more applicable and important with the degree of novelty being faced. Novelty is also linked to the degree of knowledge in a given situation and is something you will consider later in this course.

SAQ2

Think about the position of AutoCAD R13, AutoCAD LT, Mechanical Desktop and Workcenter and place them on the spectrum in [Figure 3](#).

Answer

My positioning of the four products is as shown in [Figure 4](#).

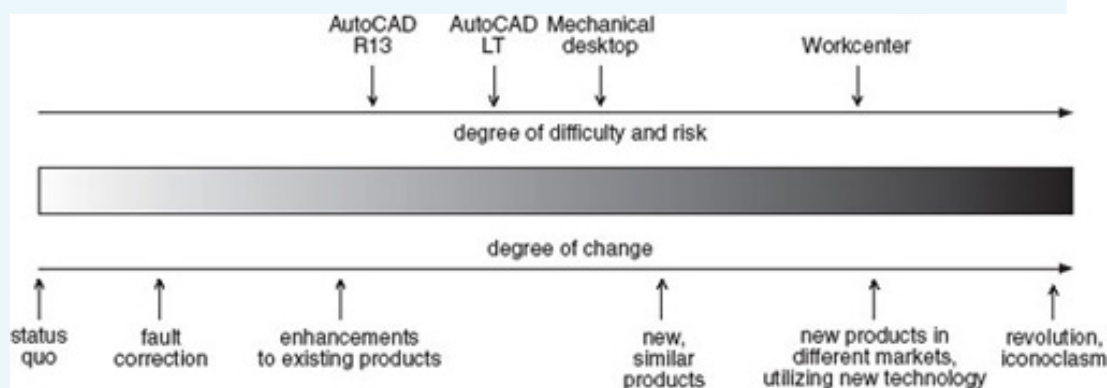


Figure 4 The positioning of the four products

Question 1

What would you conclude from the Workcenter case and your analysis in SAQ2?

Answer

The main thing that strikes me is that perhaps it is a mistake to assume that a single development process – the PDP – could apply equally to different classifications of project involving different degrees of change and risk.

1.4 Example 2: The Bridge of Sighs (and Wobbles)

The second example of a systems failure is the Millennium Bridge across the Thames linking the St Paul's area of the City of London on the north side of the river with a new cultural centre emerging in Southwark on the South Bank.

The bridge was to be solely for pedestrians, and was an idea first put forward by David Bell, who was at that time Editor of the influential *Financial Times* newspaper. The paper sponsored an international design competition which was won by a consortium of the architects Foster and Partners under the leadership of Lord Foster, Sir Anthony Caro, the distinguished sculptor and Ove Arup, the well-known firm of consulting engineers. All three members of the consortium had international reputations for innovation and the quality of their artistic and design capability. Foster and Caro's concept for the bridge was daring and imaginative. Foster, inspired by the light beam of legendary superhero Flash Gordon, envisioned a 'beautiful blade of light'.

The bridge was one of a number of similar projects that had been stimulated by the success of Santiago Calatrava's Alamillo Bridge built for Expo'92 in Seville and the Erasmus Bridge, opened in 1996 in Rotterdam and designed by the architects UN Studio. These projects demonstrated that bridges could be potent symbols of a particular time and location and could engage the local community. Caroline Bos, co-founder of UN Studio, recalled, 'Designing the Erasmus was an incredible experience for us because the local community was so excited about it.' (*Financial Times*, 3 June 2000)

These bridges made use of new technology and materials to support innovative designs that were intended to support growing cultural tourism. The footbridge across the River Tyne at Gateshead uses a new pivoting mechanism that allows it to swing open to enable ships to pass underneath. Roger Ridsdill-Smith, an Ove Arup associate, commenting that the design of the Millennium Bridge could not have been attempted ten years previously, stated, 'An engineer then would have understood the physics of how the bridge stands up but wouldn't have had the confidence to do it. High powered computers have given us that confidence by allowing us to test our calculations. The structure is so fine that, when people stand on the bridge, they'll feel as if they're hovering over the water.' The importance of the bridge to tourism was emphasised by Lord Foster: 'When a foreign visitor comes to London, they will remember buildings like St Paul's or Tate Modern, but also the experience of moving around. Crossing the Millennium Bridge – devoid of cars, buses and trucks – will be one of their most vivid memories.' (*Financial Times*, 3 June 2000)

The 'beautiful blade of light' was to be the first bridge constructed across the Thames for over a hundred years, since the building of Tower Bridge in 1894. The design was not without some criticism, most notably from the Swiss architects of the Tate Modern

conversion, Herzog & de Meuron (*Financial Times*, 10 May 2000). An impression of the design of the bridge can be gained from the photographs in [Figure 5](#) and [Figure 6](#).

Work was launched on the sleek steel structure by the Deputy Prime Minister, John Prescott, at a ceremony held on 28 April 1999. The construction work was to be undertaken by a joint venture team of Sir Robert McAlpine and the Danish specialist bridge-builders Monberg Thorsen (*The Times*, 28 April 1999). The opening was predicted to be April 2000 and the estimated cost was £15 million. This was a considerable increase on the original estimate of £9 million. Most of the funding for the project came from the Millennium Commission, the Corporation of London (through the Bridge House Estates Trust) and the Hong Kong and Shanghai Banking Corporation (HSBC).



Figure 5 The Millennium Bridge looking from the South Bank of the Thames towards the City of London and St Paul's Cathedral



Figure 6 Close-up of the bridge's structure showing suspension cables

The 330-metre structure was designed for pedestrian use only. The shallow suspension structure is supported on two piers, with a central span of 144 metres. The weight of the bridge is carried by four 120-millimetre-diameter cables on either side attached to the deck by arms and anchored with abutments on the banks. The cables and the deck of the bridge dip by 2.3 metres over the central span. The design had to comply with several requirements. First it had to allow the navigation of a variety of ships up and down the river. At present the largest of these is the 2000-tonne *Tracey Bennet*, a commercial sand and aggregate carrier, which is 55 metres long. The design team undertook extensive studies of the effects that the impact of such a vessel would have on the bridge. The second consideration was that of stiffness.

The Institution of Civil Engineers' *Dynamics Design and Practice Guide* (Maguire and Wyatt, 2002) points out that 'dynamics is a far more important subject to civil and structural engineers than it used to be, because structures have become lighter, members (structural elements) more slender. These changes have increased amplitude of vibration and moved frequencies into bands that are more awkward to deal with and more easily perceived by users.' This consideration is particularly important for footbridges since 'most walkers on a sensitive bridge instinctively adjust their pace to the response in a way that maximises excitation.' (Maguire and Wyatt, 2002). This well-known phenomenon is the reason that marching soldiers are ordered to break step when crossing a bridge.

With these points in mind,

The original design assumptions regarding dynamic loading by pedestrians were checked. The guidance in the relevant UK bridge design codes had been followed, with additional input from overseas codes and non-statutory documents. The codes advise on the level of force to be assumed in design, and the amount of movement which is acceptable to users of a bridge.

The UK bridge codes require only vertical excitation to be considered. In the original design all critical vertical modes of vibration were examined. The size of the applied

loading recommended in the bridge design code was increased by 33 per cent to give the design additional robustness. The original design also considered loading due to groups of vandals deliberately attempting to excite the bridge.

Extensive additional research was carried out during the original design, including database searches on suspension bridges.

(Arup, 2000a)

However, perhaps the consideration that was at the forefront of the bridge designers' minds was the need for the structure to be both innovative and aesthetically bold. In this Lord Foster and Sir Anthony Caro did not have it all their own way. Sir Anthony blamed planners for interfering with the design and said that they had 'cut down and trampled on' innovative ideas (Yahoo, 2000).

The bridge did not open in April 2000 as originally planned; engineering problems and a strike at the Finnish company supplying some of the steel members delayed completion. The final cost of £18.2 million (*Financial Times*, 11 July 2000) had risen by £3 million in the year of construction and had doubled the original estimate of £9 million. The Queen performed an opening ceremony on 10 May 2000 but was unable to walk across the bridge as it was still in an unfinished state.

To celebrate the opening of the bridge to the public on Saturday 10 June, 100,000 people crossed it in a charity walk in aid of the Save the Children fund. Problems were immediately apparent and police closed the bridge as it began to sway alarmingly in only a light wind. A similar problem had occurred earlier in the year when a pedestrian bridge in Paris, linking the *Quay d'Orsay* with the *Tuileries* gardens, was closed after the French Minister for Culture, Christine Trautmann, was upset by similar movements (*Financial Times*, 12 June 2000).

The Millennium Bridge remained open the following day but only 150 people were allowed to cross at one time. Even so, some visitors complained that they could not walk in a straight line, and of feeling seasick. Others enjoyed the experience. Kay Clapton from Kent stated, 'If they repair the bridge it won't be the same. The thrill of walking across the Thames as the bridge is swaying is amazing.' (*Financial Times*, 12 June 2000). In spite of reactions such as this from the more adventurous, the decision was taken to close the bridge. Ove Arup explained:

The bridge was open for three days, between Saturday 10 June and Monday 12 June. At times it was very crowded, with some 2000 people along its length. An estimated 80,000–100,000 people crossed on the opening Saturday. When the bridge was crowded, the south and centre spans suffered lateral vibrations large enough to cause pedestrians to stop walking or to hold onto the balustrades to regain their balance. The movement of the south span, between Bankside and the first river pier, was a combination of horizontal and torsional (twisting) oscillations. Observations on the day and studies of video footage show up to 50 mm movement, depending on the number of people walking. The frequency of the movement was about 0.77 cycles/second.

The centre span moved by up to 70 mm at a frequency of 0.95 cycles/second, mainly horizontally. This part of the bridge was observed to oscillate when occupied by more than about 200 people. The north span did not move substantially.

It was decided to control the number of people present on the bridge from noon on the Saturday onwards. The concern was the safety of individuals, rather than any risk of structural failure of the bridge itself.

The bridge was closed on 12 June pending investigation into the cause of the unexpected movements.

(Arup, 2000a)

There followed weeks of intensive investigation. The Building Research Establishment provided specialist shaking equipment that was used to replicate the forces acting on the bridge. A group of 100 Ove Arup employees crossed the bridge in an attempt to replicate the behaviour of groups of people in response to instability. Vertical and horizontal loading were monitored at key points by the Transport Research Laboratory during the tests, which were filmed from four locations.

As a result of these tests the engineers went back to the model that they had used to check the original design. The work had been undertaken following best practice principles, which require the design and analysis to be carried out independently by two separate teams. The results of the work of these teams was then checked. In addition, the work was validated by an independent firm, who conducted their own analysis. All this work was replicated using data from the tests.

The results of this review and comparison show that, apart from the unexpected movement, the bridge is reacting as predicted. There are some marginal – and explicable – differences in the figures but none represents a significant departure from the model, and none could explain the unexpected movement.

(Arup, 2000b)

The problem of the wobbling bridges seemed to be caused by *people walking across it!* At one level of analysis this assessment was correct. However, the potential problem caused to the structure of bridges by soldiers marching in step was well known. The problems experienced by the Millennium Bridge had a different origin, to do with *induced* walking in step and the bridge structure. As groups walk across the bridge the natural cadence of their steps may coincide with the horizontal frequencies of the bridge structure. If the movement brought about by such a coincidence becomes noticeable to those crossing they will tend to adjust their walking pattern in order to feel more stable and, inadvertently, respond in a way that 'maximises excitation', as the Institution of Civil Engineers' *Dynamics Design and Practice Guide* (Maguire and Wyatt, 2002) points out. According to an expert, Professor Jonathon Wood, who had worked on a similar problem on the Severn Bridge, 'The Millennium Bridge has almost no lateral stiffness and virtually no torsional stiffness.' *The Times*, 31 October 2000)

Midway through September 2000 Ove Arup was in a position to report on its investigations. It confirmed that the wobble had been caused not by the lack of lateral and torsional stiffness in the original design but by 'large groups of pedestrians [who] had appeared to be stepping in time with the motion of the bridge.' On its website, Arup notes:

A programme of research was undertaken during the summer of 2000. A solution was then developed using the results of these tests. Arup has warned other bridge designers of their findings and the British Standard code of bridge loading is being updated to cover the phenomenon, now becoming referred to as Synchronous Lateral Excitation.

The research indicated that the movement was caused by the sideways loads we generate when walking. Chance correlation of our footsteps when we walk in a crowd generated slight sideways movements of the bridge. It then became more comfortable for people to walk in synchronisation with the bridge movement.

This instinctive behavior ensures that the sideways forces we exert match the resonant frequency of the bridge, and are timed such as to increase the motion of the bridge. As the magnitude of the motion increases, the total sideways force increases and we become more correlated.

The sway movement is not specific to the Millennium Bridge. The same excessive sway movement could occur on other bridges, future or existing, with a lateral frequency under ~ 1.3 Hz and with a sufficient number of pedestrians.

During the investigations Arup discovered that other bridges with completely different structures to the Millennium Bridge have swayed sideways when crowded, for example the Auckland Harbour Road Bridge during a demonstration in 1975 [...]. These cases have not been widely published and as a result the phenomenon has not become known to practicing bridge engineers.

(Arup, 2005)

The engineers considered four solutions. Two of these were rejected early in the analysis. These were solutions to *limit the number of people using the bridge* at any one time, or to *modify walking patterns* by introducing obstructions. These options were considered to be undesirable and a last resort to be adopted if no other solution could be found.

The engineering team estimated that stiffening the structure to alter the 'natural frequency range of the bridge sway modes away from the frequency range of lateral forces from pedestrians' would require a nine-fold increase in lateral stiffness in the centre span of the bridge. As a result, heavy bracing would be needed. This solution was rejected because it would have required, in addition, conventional dampers to be fitted. The total weight of such a scheme would have meant further modifications to the bridge's structure and foundations. It would also have altered the appearance of the bridge.

The solution adopted, subject to a satisfactory trial, was to fit two sorts of damper to absorb the vibrations created in the structure. Viscous dampers, which operate like car shock absorbers, restrict the flow of a viscous liquid contained within the device. These were to be fitted between the existing structure and some additional steel bracing. Tuned mass dampers were also to be used to provide additional control of both vertical and horizontal movements. Preserving the appearance of the bridge was of special importance to the members of the consortium. Ove Arup (Arup, 2000a) stated:

Protection of the elegance of the bridge has been a key aim. The design solution has evolved in weekly meetings with Foster and Partners and discussed in detail with Sir Anthony Caro and Lord Foster. [...] The additional structure and dampers under the bridge deck do not affect the side elevation since they are located above the level of the transverse arms and therefore behind the deck edge tubes. The structure which will be visible on the bridge elevation will be:

- The diagonal viscous dampers at the piers. These are located in the plane between the cables and the edge tubes and extend two bays along on each side of the pier.
- The two viscous dampers on each side of the south abutment ramp. These are arranged as an inverted 'V' and are located on each corner of the ramp junction where it splits in two and turns parallel to the river bank.

The underside of the bridge is an important feature of the design. One of the main views of the bridge is from each bank and from river boats, where the soffit is plainly visible. The chevron shaped bracings have been designed with this in mind and are aligned with the bridge centreline and spaced at regular intervals. Similarly, the

tuned mass dampers are arranged regularly and placed between the underside of the deck and the top of the transverse arms.

The proposed solution was tested using the model, the analysis confirming that the dampers would cure the wobble or at least limit movement to 'acceptable levels'. Further modelling was undertaken to test the solution in exceptional circumstances. Finally, tests were to be carried out on individual components and a prototype test carried out to determine the dynamic behaviour of the bridge with a set of dampers fitted.

In late November 2001 work finally started on the modifications. Ove Arup paid a reported £250,000 to have two viscous dampers and one tuned mass damper fitted to the central span of the bridge. These paved the way for the full solution, which was estimated to cost £5 million.

SAQ 3

Make brief notes in answer to the following questions:

1. What was the problem that occurred at the brief abortive opening of the bridge in June 2000?
2. What work was undertaken to define the problem in detail?
3. What measures of performance were used evaluate the proposed solutions?
4. What solutions were considered?
5. How were the solutions evaluated?

Answer

1. The bridge wobbled to an extent that discomforted some users and might possibly have been hazardous.
2. Extensive tests were carried out in an effort to understand the problem and to identify its cause. A shaking machine was brought in and groups of Ove Arup employees walked across the bridge in an attempt to replicate the problem. The movement of the bridge was monitored and filmed using video cameras. The results of the tests were fed into the model that had been used to validate the original design.
Advice was taken from experts in the field and the behaviour of other bridges was researched.
3. First, the proposed solution had to solve the wobble problem to reduce the movement of the bridge to an acceptable level. Second, it had to avoid degrading the appearance of the structure.
4. Four solutions were considered:
 - restricting the number of people crossing the bridge at any one time;
 - breaking up the stride pattern of people crossing;
 - stiffening the bridge;
 - damping the bridge's movement.
5. Two of the possible solutions were considered to be impractical. The stiffening solution was rejected on engineering and aesthetic grounds. The damping solution was tested using the model. Individual components were tested. Finally a prototype was installed on the bridge and was tested.

Multiple-cause and sign-graph diagramming are techniques that can be used to identify and understand the relationships between the various elements in a situation.

Recently I was discussing with a group of colleagues the importance of reducing the time that it takes to change over from one job to another on a factory floor. In a 'semi-brainstorming' mode we quickly drew the diagram shown in [Figure 7](#) on a flipchart.

The arrows connecting one element in the diagram with another should be read as meaning 'influences', 'affects' or 'contributes to' rather than 'causes'. For example, the changeover time affects the available capacity of machines in the costs of holding system. We added '+' and '-' signs to indicate the 'direction' of the influence of one element on another. The '-' sign indicates that, for example, as changeover time *increases*, the capacity available for production *decreases*. The changes in the two elements work in opposite 'directions' to one another, so it would also be true that an *increase* in changeover time would be accompanied by a *decrease* in the capacity available for production. The '+' is used to designate an influence where movements in two elements work in the same direction, so that if inventory increases, the value of work in progress (WIP) will also rise. Similarly, if inventory falls, the value of WIP will fall.

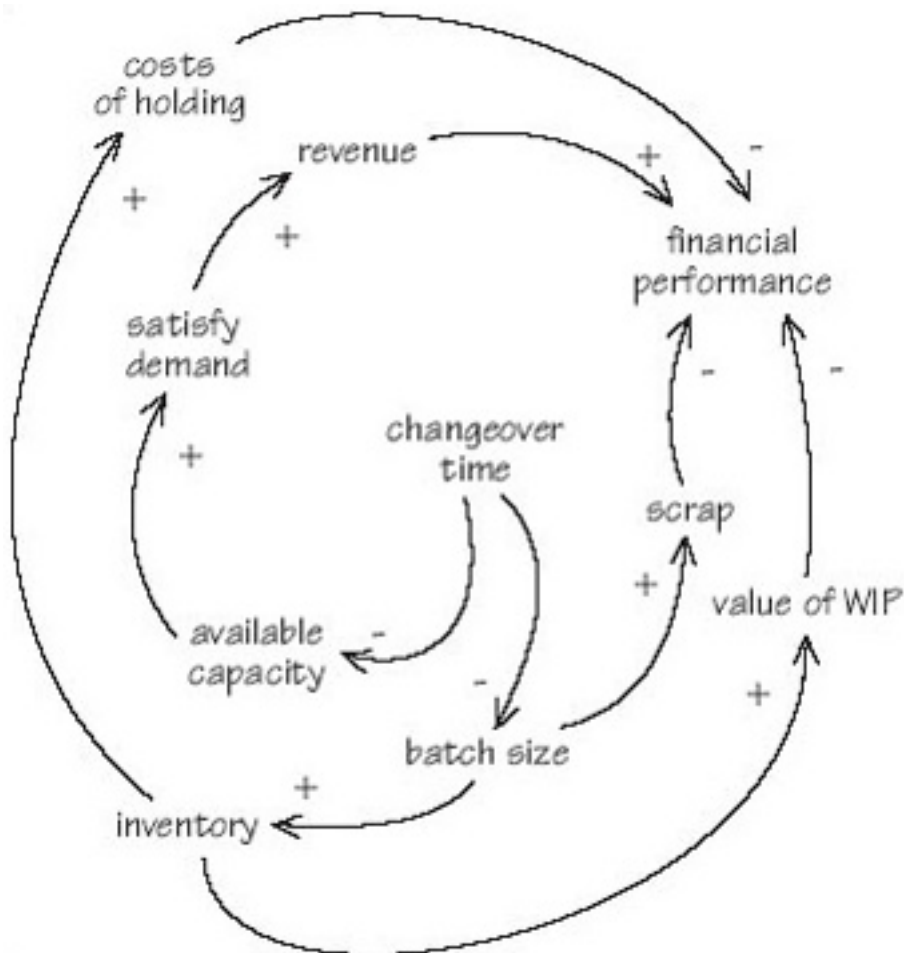


Figure 7 The importance of job changeover times

I said earlier that [Figure 7](#) was the result of a semi-brainstorming session. As such it was an initial statement that needed considerable refinement. It was certainly not the 'finished product' but played a valuable part in getting the team started.

SAQ 4

Draw a multiple-cause diagram that would provide an initial understanding of the reasons for the wobble problem on the Millennium Bridge.

Answer

My diagram looked like [Figure 8](#).

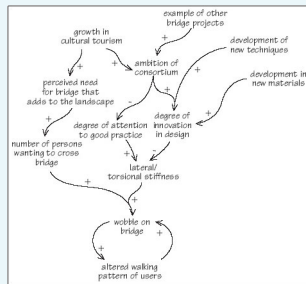


Figure 8 Reasons for the Millennium Bridge wobble

The problems of designing software that works have been apparent for forty years and those associated with designing and building bridges for thousands of years. This collective experience has not stopped designers and engineers making mistakes, which are often expensive in terms of time and resources. Autodesk spent millions of dollars developing Workcenter but abandoned the product soon after its launch. The original budget for the Millennium Bridge was £9 million; the final cost, including modifications, was approximately £23 million, approaching three times the first estimate. The opening of the bridge was planned to coincide with the launch of the Tate Modern in April 2000 but failed to meet that date.

In both cases the designers and engineers were being innovative. They were trying to operate towards the right-hand end of the spectrum of change illustrated in [Figure 3](#). This makes the endeavour risky and, therefore, mistakes more likely. This does not mean that we should avoid risk. Life would be a lot duller if risk avoidance were always the norm. In the two examples, and with the benefit of hindsight, it is clear that both sets of designers failed to conceive their projects appropriately. The software engineers at Autodesk thought that they were building a software package rather than a product that would have to include both software and consultancy. Lord Foster and Sir Anthony Caro wanted to design a *beautiful* bridge. In doing so they lost sight of its function. In both cases a more holistic methodology might have prevented the problem that occurred. Systems engineering aims to provide such a methodology.

1.5 Increasing complication, complexity and risk: the underlying relationship

[Figure 3](#) showed five commonly encountered problems of effecting different types of change. These are notionally located on a spectrum of change that ranges from no change at all, to complete revolution. The relationship suggested on the figure is that as the degree of change – represented by the different types of problem – increases so, too, do difficulty and risk. Each of the five problems of effecting change can be regarded as a

gap between an existing situation and an alternative, desired or preferable situation. To close these gaps, be they the correction of faults or the design and implementation of a completely new, innovative system, requires the deployment and consumption of resources. These resources may be a mixture people, materials, equipment, objects, and information. These two characteristics of change – the nature of the change problem and the use of resources – can be used as the basis for developing the picture shown in [Figure 3](#) into a more complete model of change, and taken together represent the certainty of outcome of a change project.

[Figure 9](#) shows change problems divided into three categories; simple, complicated and complex. The two dimensions of the problem of effecting change, knowing what is required and knowing how to achieve what is required are shown as two axes that each run from high to low – from complete knowledge and certainty to 'haven't a clue'. Change problems are rendered (relatively) *simple* by a high degree of knowledge of what needs to be done and how to do it. I have termed the dominant form of knowledge required to address this type of problem as 'craft knowledge', since it is the product of a learning that is essentially experiential in character, being a result of meeting and tackling successfully similar problems in the past. This craft knowledge has been termed as 'tacit' and may be embedded in individuals or in the organisation itself in informal rules and procedures. In such situations certainty of outcome is high.

As uncertainty increases the change problem becomes more complicated and less amenable to solution through the application of tacit knowledge. Knowledge of what is needed or how to achieve what is required, or both, is less certain. In such situations those involved are likely to fall back on formal knowledge, either of first principles or that which has been embedded in formal rules and procedures. These *complicated* change tasks are often solved by the application of traditional engineering knowledge.

The third type of change problem shown in [Figure 9](#) has high uncertainty as it is *complex*. Traditional forms of engineering knowledge no longer suffice and it is in application to this type of problem that systems engineering knowledge comes into its own. This type of knowledge is both 'systemic' and 'systematic'. It is systemic because, being based on the systems principle of 'holism', it views the change problem as a whole, resisting the inclination to see it from the perspective of a particular function or discipline. It is 'systematic' because it embodies rational frameworks and approaches that reduce uncertainty.

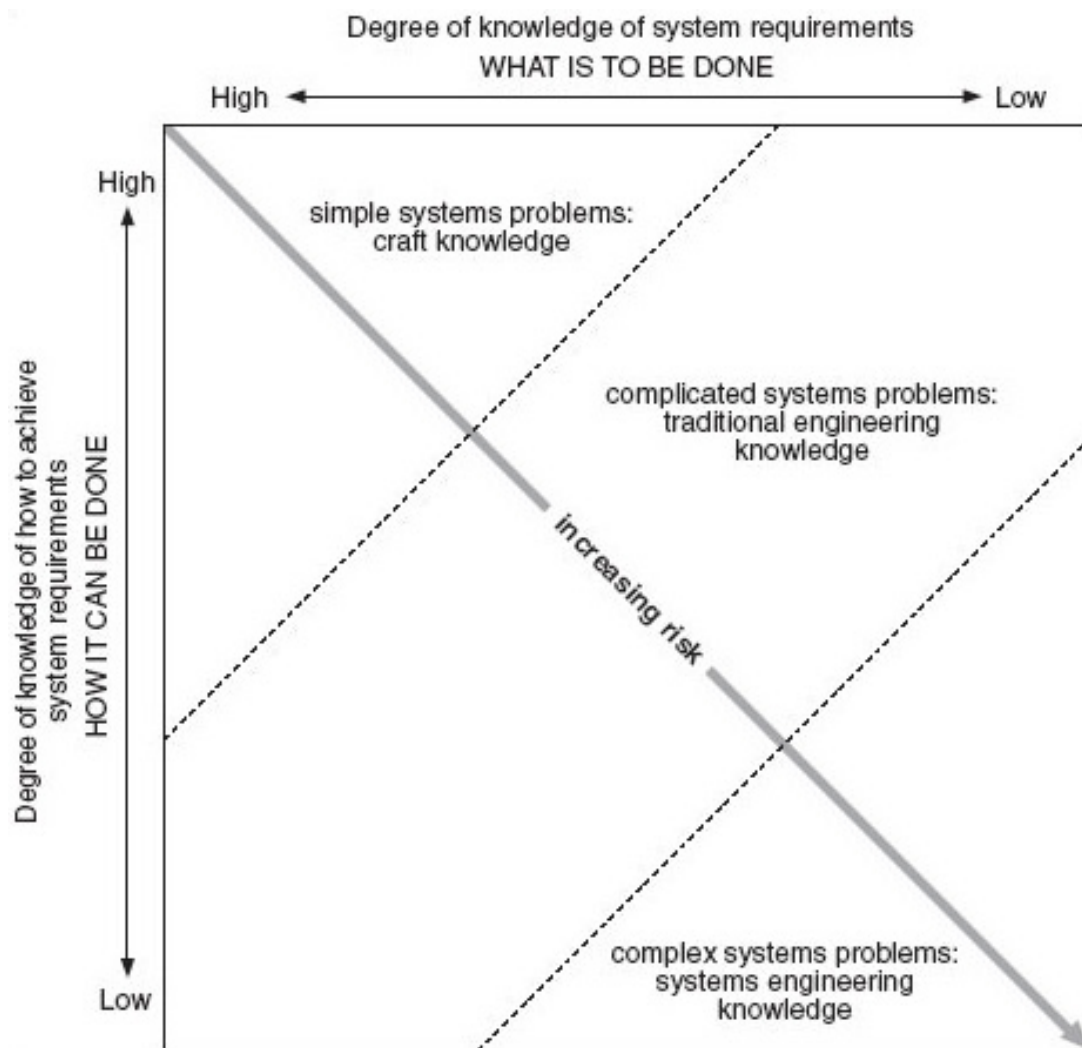


Figure 9 A model of simplicity, complication and complexity of systems problem related to type of knowledge

Question 2

Do you consider that the degree of knowledge of what is to be done and how it can be achieved explains the increasing risk completely?

.....

Answer

This point gave rise to a fierce debate in the original Course Team. The External Assessor argued strongly that there are other factors than the degree of knowledge that contribute to complexity such as the increased coupling of systems and the strength and breadth of their interaction. These give rise to stronger emergent behaviour. While agreeing with this point, others stated that it was the lack of knowledge and, therefore, the failure to predict these factors that gave rise to increasing risk.

An example of the three types of change problem will help to illustrate the model.

Suppose that I want to extend the electrical wiring in my house into the garden to run some lights and a water feature. Designing an extension of this type is well within the capability of a competent electrician, and I am confident that the lights and water feature can be got to work satisfactorily and safely. Craft knowledge of a readily available kind is required to deal successfully with this simple problem.

A more complicated problem is consequent on a decision to install a new security system for the house. This requires more specialist knowledge than a simple extension to existing wiring and is a more difficult design task. I am likely to employ a firm that specialises in this type of project, which may also involve, in supporting roles, other areas of knowledge such as glaziers, plasterers and so on.

The complex level of change problem can be illustrated by the decision of a building company to design a dwelling with a fully automated control system. Since, as yet, only limited prototypes of this type of accommodation have been developed, the requirements and functionality of such a system are uncertain. Equally vague are the domains of knowledge that would be needed to design and implement such a system successfully. As a result, the outcomes of a project to design a 'house automation system' are highly uncertain.

Four observations can be made about the change problem model presented in [Figure 9](#). Uncertainty increases with the degree of turbulence in the environment of the change problem. This turbulence may be associated with change in:

- underlying technologies, either those embodied in the product or service in question or those that are to do with how to achieve what is required
- the business or competitive environment
- the political environment
- the social environment
- the economic environment.

The existing knowledge base of individuals and organisations will bias their perception of a problem and how it can be solved. The organisation may suffer from unconscious incompetence, not being aware of what it does not know. These two factors, environmental change and perception of the nature of the problem, increase the degree of uncertainty that is associated with a need for change and, consequentially, with its riskiness.

Risk can be denned as the probability of an unexpected outcome. Naturally enough we have an asymmetrical attitude to unexpectedness. We don't mind positive unexpected outcomes but want to avoid nasty ones and their consequences. Our tendency is, therefore, to be risk averse, and only if we are offered a greater return for doing so will we take on extra risk:

- those undertaking dangerous sports are compensated by the psychological return that they provide, or the social status that participation confers
- punters on a horse race are offered better odds on outsiders than on the favourite
- motor insurance companies want bigger premiums from drivers with a poor claims history
- investors in the stock market demand greater returns from shares that are more volatile than the average of the market as a whole.

As a consequence, though it may be tempting to do so, businesses which undertake only safe forms of change, those that fall within their comfort zone, will not do better than the

average. Taking on risk is uncomfortable but necessary because it brings with it greater financial returns and increased knowledge and learning. Systems engineering is a way of reducing the inherent riskiness of the new and complex.

In the remainder of this section of the course I will discuss the issues for systems engineering associated with the topics of *simplicity*, *complication*, and *complexity*.

Following the model in [Figure 9](#), these closely related and, in some instances, overlapping topics will be examined in relation to the difficulty that they create for a systems engineer in terms of what he or she has to do, and the way that the work is performed.

1.6 Increasing complication, complexity and risk: mystery and mechanics

The winter of 1665/66 must have been exceptionally harrowing for the inhabitants of England. Along with the winter weather, the country suffered an outbreak of the plague. A minor effect of this was a decision by Trinity College Cambridge to close its doors. One of those affected by this decision was a young Fellow, Isaac Newton, who returned home to spend the winter in the Lincolnshire rectory in which he had been brought up.

Isolated in the bleak fens and without college high table and the conviviality of the other Fellows to distract him, he found himself at a loose end, so the 22-year-old Newton buckled down and during the next 12 months:

- solved the binomial theorem
- invented calculus
- discovered the universal law of gravitation
- developed a theory of colour.

Eventually the threat of the plague lessened and Newton returned to Trinity, where he was elected Lucasian Professor of Mathematics. The work that he did during the 1665/66 winter became the basis for *Philosophiae Naturalis Principia Mathematica* (The Mathematical Principles of Natural Philosophy), which was first published in 1687.

The importance of Newton's work cannot be overestimated, and it is no exaggeration to regard 1665/66 as the beginning of the modern world. The mysterious, magical world of the Middle Ages was replaced by one amenable to rational analysis. Explanation based on myth, magic or the unknowable will of a divinity gave way to observation, calculation and the operation of universal laws. This approach was so successful that 250 years later the French mathematician Henri Poincaré (1854–1912) stated:

If we know exactly the laws of nature and the situation of the universe at the initial moment, we would predict exactly the situation of that same universe at a succeeding moment.

(Poincaré, 1995 [1903])

The achievement of Newton, and others who built on his work, was to provide ways of understanding relationships and interactions in the physical, observable world. In doing so they reduced its complexity to mere complication at worst and simplicity at best.

As suggested earlier in this section, simplicity, complication and complexity are closely related to perception, understanding and the existing knowledge base. If we are faced with a problem that we do not fully understand or one that we cannot see how to solve, we label it 'complex'. Effectively, we are saying that there is an unknown area that needs to be

explored, and a way of dealing with it established. A close conceptual relation of complexity is *complication*.

The wristwatch that I habitually wear happens to have a glass back through which its mechanism can be viewed, as shown in [Figure 10](#). It's an interesting world inside the watch case, with lots of tiny parts interacting with one another. It is complicated but not complex. There is no 'unknown' element in the nature of the outputs of the watch or how the mechanism achieves them. Although personally I couldn't construct a watch, there are plenty of people with the necessary skills. It's a 'known problem', albeit a complicated, tricky one.



Figure 10 The wristwatch mechanism

My watch is an illustration of the physical world of objects governed by Newtonian physics and for which, therefore, we have good explanatory models. There are, however, three other 'worlds' for which we do not, as yet, have models of equal stature.

[Figure 11](#) shows our level of explanatory confidence as a function of three worlds – the 'sub-physical' world of quantum physics, the world of physical objects governed by Newtonian mechanics and a 'supra-physical' world of complex systems and which includes a fourth world of human activity systems. In both the sub- and supra-physical worlds there is considerably less success of explanation and therefore greater inherent complexity.

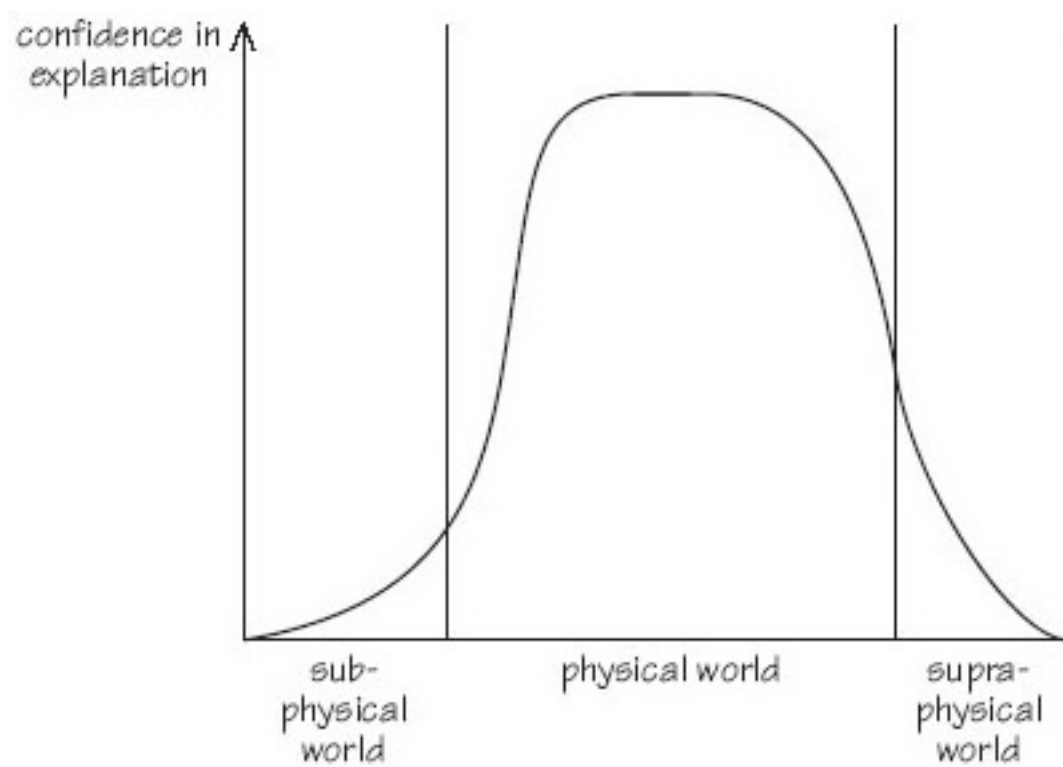


Figure 11 Explanatory confidence in three worlds

In 1927, the German particle physicist Walter Heisenberg (1901–1976) put forward the view that at a subatomic level it was possible to determine either the location of a particle or its vector, but not both. In order to study the behaviour of subatomic particles it is necessary to bounce other subatomic particles off them or to get them to collide with other subatomic particles. Either of these two actions destroys what was happening and so leaves it a mystery. Heisenberg's uncertainty principle states that what happens down in the depths of the subatomic world is *unknowable*.

In 1968, the German theoretical biologist Ludwig von Bertalanffy published *General System Theory* (von Bertalanffy, 1968). Although elements of this work had precursors, von Bertalanffy's work was essentially the basis of academic interest in 'systems' as a subject. The conceptual basis of systems is discussed in more detail later but one of its cornerstones – *emergent properties* – is relevant here. This concept states that the properties and behaviour of a system cannot be deduced from studying the properties or behaviours of its elements in isolation. At one level this principle hardly rises above the banal. Everything, be it a physical object or conceptual system, exhibits properties and behaviours that result from the interaction of its constituent elements and which, therefore, are not to be found in those elements in isolation. There are, I believe, no exceptions to this statement. This means that the possession of emergent properties cannot be regarded as a distinguishing feature of a system. However, it is often the case that systems, even simple ones, exhibit behaviours that are unexpected and which surprise their designers, users or observers. Sometimes these behaviours could have been foreseen, but through oversight or negligence were not considered during the design phase of the system. Of equal interest to these preventable emergent properties are those that could not have been foreseen and which are genuinely unexpected. There are external and internal reasons for the occurrence of these. This point will be examined in more detail in Section 3.

Externally caused emergence occurs when a system reacts to its environment in a way that could not have been predicted. There are two origins of unexpected externally caused emergence.

1. The system has not been designed to be robust against variation in part of its environment. For example, part-way through the construction of the light-railway system serving London's Docklands area an announcement was made of the massive office development at Canary Wharf. On its own this project completely negated all the carefully calculated predictions of traffic for the new railway. To compound the problem the Canary Wharf announcement was made when the construction of the railway, its rolling stock and associated traffic management systems were all well advanced. It was thought that nothing could be done to accommodate the traffic that the Canary Wharf development was expected to generate but, in the event, the railway has coped remarkably well. Emergent properties do not always mean failure.
2. The occurrence of a new element in the system's environment. The example of the Docklands Light Railway illustrated an unexpected variation in one of the important parameters used as the basis for the railway's design. Sometimes, however, a new factor will occur in the environment. The more complex the system, the longer (all other things being equal) the design, development and implementation processes take and therefore the more likely that unpredictable factors affecting the system will occur in the environment. The Iridium system was conceived as providing worldwide telecommunications through a network of geostationary satellites. Problems with the launch vehicles and the performance of the satellites themselves delayed the project, which was 12 years in development. In the meantime the interconnection of networks of terrestrial systems had overtaken the concept. Iridium declared itself bankrupt in August 1999 but may be resurrected as a system for specialist communication.

Internally generated emergence occurs when the elements of the system interact with each other in an unpredicted way. A potent source of this type of emergence is created by the behaviour of humans (most often, but other sentient creatures too) within the system. Once again, the Millennium Bridge provides an example of this. If only the people crossing the bridge had not perversely attempted to compensate for its lateral movement everything would have been fine and the 'blade of light' would have remained unsullied by dampers and struts. Emergence can also arise from the unforeseen interactions between the elements of the system and its environment.

Because we do not know *everything* which is salient when that knowledge is required, the often unexpected, unpredictable character of emergence means that it remains mysterious, adding to the difficulties of undertaking a complex systems engineering task.

1.7 Increasing complication, complexity and risk: a spectrum of systems intractability

Summarising the discussion in the previous two sections, [Figure 12](#) shows what might be termed 'a spectrum of systems intractability'. At one end of the scale are simple systems. These are easily understandable and their design and development (relatively) unproblematic. The way in which the various elements in the system fit and work together is clear. Outputs and behaviours are predictable. An example of a simple system is the table shown in [Figure 13](#).

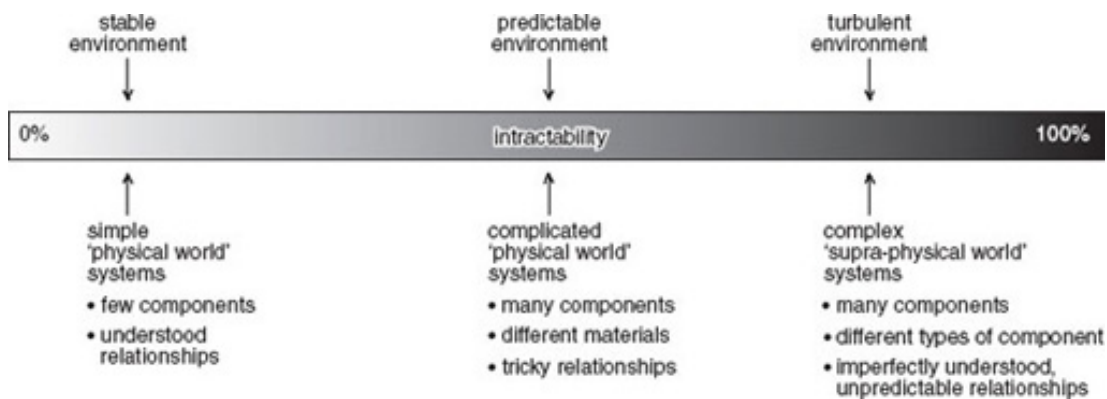


Figure 12 A spectrum of systems intractability



Figure 13 A simple, physical world system

The table is constructed from 42 parts made from three types of material and manufactured using different processes:

- top – made of plywood
- veneer for surface and edges of top (wood)
- four plywood legs consisting of two glued components
- twenty-six cross-headed screws
- two metal fixings for securing one table to another.

While the table can be regarded as a system itself, it may also be useful to think of it as part of the 'presentation' room system shown in [Figure 14](#).



Figure 14 The presentation room system

The number of elements in this system has increased enormously. There are, for example, 12 tables in the room. Each of the 20 chairs has 53 separate components but an additional level of complication is introduced by the personal computer, the projector, the hardcopy projection device and the links between them. The relationship between the components of the table is mechanical, that between the tables and chairs spatial, but that between the projector and the personal computer is both physical (through the wires) and informational, expressed through various layers of software. If each table can be viewed as a simple system, the presentation room as a whole has become complicated. It is not complex since its elements can be understood and the relationships between them defined. The behaviour of the system shown in [Figure 14](#) is predictable since it is static.

[Figure 15](#) shows the presentation room system with the addition of people to produce what might be labelled a 'learning system'. The addition of people to the presentation room turns it from a static system to a dynamic one, and its behaviours from predictable to unpredictable. The result is to push the system towards the right in the spectrum shown in [Figure 12](#). It has become complex rather than being merely complicated.



Figure 15 The learning system

As the examples illustrated in [Figures 9](#) and [13–15](#) suggest, the factors that increase complication, and hence complexity, are as follows:

1. The number of separate components in the system. A greater number of components leads to increased complication.

2. The variety of components in the system. The component parts in the 'table system' are made from only three different types of material.
3. The nature of the relationships between the components. In the table system the relationship between the parts was mechanical and chemical (the adherence of the veneers to the plywood top), whereas in the learning system the types of relationship are, in addition, electrical, electronic, informational and psychological.
4. The degree of coupling between the parts of the system. The various parts of the table were tightly coupled one to another, but the coupling of some of the components of the learning system is a great deal more loose. The way that, say, the application software in the personal computer interacts with its operating system may not be fully denned, leading to the (all too familiar) system crashes; the interaction of the computer with the projection equipment often has a degree of uncertainty. However, the elements in the system that exhibit the loosest coupling are often the human beings in their interaction with the non-sentient elements of the system and their relationship with one another. For example, the replacement of human workers with automation will increase the coupling of the system.
5. The degree to which the operation of the system is dynamic. In a static system, like the table, the relationship of the parts is necessarily fixed. The functioning of the table (as a table rather than, say, a barrier) demands that the relationship of its parts is stable and that it exists in a certain aspect to the floor of the room. The relationship between factors 1–3 listed above and complexity is positive.
6. The degree to which the system is robust in the face of variations in its environment. Systems maintain a degree of integrity when confronted or challenged by changes in their environment. They have a designed in or learned 'envelope of robustness'. Thus, the wristwatch shown in [Figure 10](#) proclaims that it is 'shock resistant' and 'water resistant to a depth of 30 metres'. In addition, its self-winding mechanism provides it with a power reserve of about 40 hours. The robustness of the watch was designed in; other systems – particularly those in which the human element is significant – exhibit learning, which increases robustness. A robust system is more predictable and, therefore, less complex than one which is not as robust.

Box 3 Ants at the edge of chaos

Most people would regard ants as relatively simple creatures but, as a series of experiments undertaken during the mid-1980s demonstrated, their behaviours, when viewed at a system level, can produce surprising and unexplainable outcomes.

The design of the experiments was simple. First take a nest of ants. Second, place two identical piles of food equidistant from the entrance to the nest. Third, replace grains of food that are removed from the piles so that they remain identical. The question is: once the ants are released, will they all go to one source of food, or divide themselves in some proportion between the two piles?

Since there was no reason for an individual ant (so far as is known) to prefer one food pile over the other, it might be expected that the colony would divide itself evenly, roughly half going to one pile and the remainder to the other. Each ant emerges from the nest, mentally tosses a coin, and makes for one pile or the other. Having been successful and the food pile remaining constant, the ant has no reason to change its behaviour.

However, it is known that an ant, having successfully found a source of food, will pass on the good news to others and try to persuade them to follow it by a chemical

secretion. Successful behaviour is reinforced by this means and a positive feedback loop established.

The result should be that eventually all the foraging ants are persuaded to visit just one of the food piles or that the proportions might settle down to be different from a 50:50 split, the exact ratio being established by variations in the foraging pattern.

‘In fact what was seen to take place was a completely different outcome. Even when the experiment had been running for some time, in ant terms, the proportion of the ant population visiting any one site continued to fluctuate in an apparently random fashion. The proportions averaged out at one half, but this precise outcome was hardly ever observed, and the proportion was subject to constant change. Once a large majority of ants had visited one of the sites, the outcome tended to stay reasonably stable and exhibited small variations around that proportion for some considerable time. But the majority was always eroded and the ants switched to visiting the other site. Sometimes these shifts were not only very large – from, say, an 80:20 division at one pile to the reverse outcome of 20:80 – but also rapid.’

The experiments were varied to see whether a different outcome could be induced. Different species of ants were used with no difference. To eliminate possible differences in the food sources a single pile was used. Two separate bridges were set up at identical distances from the entrance to the nest and the numbers of ants crossing each were counted. Again, the results replicated the original experiments.

The behaviour of the ants was of interest not only to biologists, and Alan Kirman, then at the European University Institute in Florence, began to look at the problem from a different view point.

‘Kirman set up a theoretical model which gives an excellent account of the observed behaviour of the seemingly perverse ants [...] An ant coming out of the nest follows one of three possibilities: it visits the food pile it previously visited; it is persuaded by a returning ant to visit the other source; or, of its own volition, it decides to try the other pile itself. And this is almost all that is required to explain the complex and seemingly baffling phenomenon of the fluctuations in the proportions of ants visiting the respective piles.’

Adapted from Ormerod (1998)

The greater the value of the factor, the more complicated and complex the system being considered or designed. In the case of what I have termed the degree of coupling, the tightness of coupling is inversely correlated with complication and complexity. The six factors combine to determine, in part, the degree to which the behaviour of the system and its outcomes are unpredictable. This, in turn, influences the extent to which the system is complex rather than complicated or simple. A degree of dynamism is shown by the watch in [Figure 10](#). In this system the functioning depends on the parts moving in relation to one another in a predetermined and predictable way. The inclusion of sentient creatures within the boundary of a system introduces a greater degree of dynamic behaviour. Even when these creatures may be regarded as relatively programmed or ‘hard-wired’ in their reaction to stimuli the dynamic behaviours that result from their interaction can be unexpected and initially unexplainable, as the example given in Box 3 suggests. However, perhaps it is our *expectation* which is the problem rather than the behaviour of systems themselves. We should anticipate variation, and as the current saying is, we should ‘Learn to expect the unexpected.’

SAQ 5

Draw a multiple-cause diagram that identifies the various relationships involved in complication and complexity.

Answer

My diagram looked like [Figure 16](#).

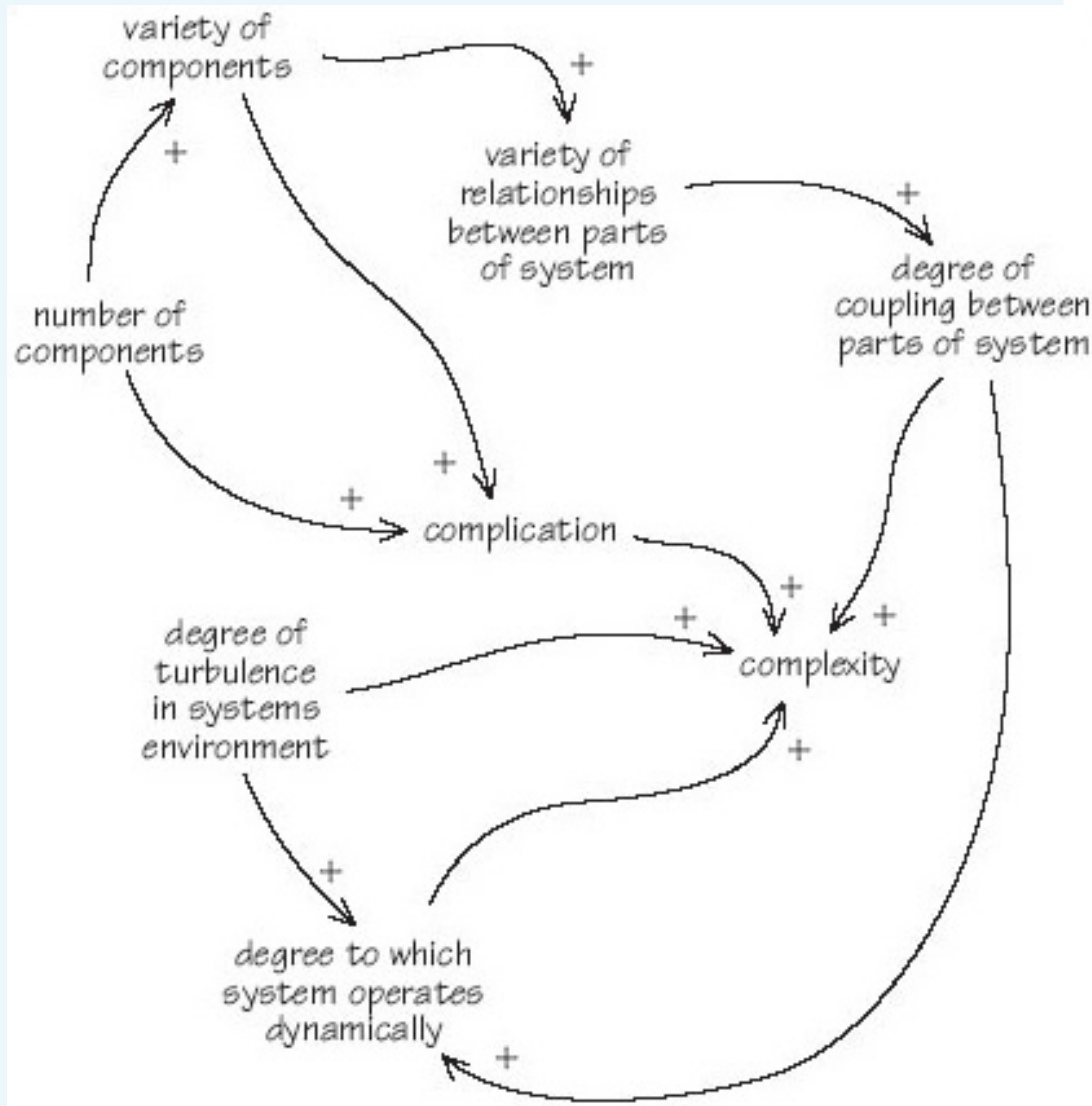


Figure 16 Factors leading to system complication and complexity

1.8 Increasing complication, complexity and risk: are systems becoming more complex?

[Figure 17](#) shows the evolution of two commonly encountered applications of systems – for personal transport and for the reproduction of recorded music. In both cases the degree of

complexity of the systems application has increased over time. One of the main reasons for this is technology push. The importance of technology can be related to the stages of the product life cycle shown in [Figure 18](#). The hypothesis behind [Figure 18](#) is that the sales of a given product or product type (such as vinyl long-playing records) will in any given market follow the characteristic sales curve illustrated. The curve is divided into the stages of introduction, growth, maturity and decline, though more elaborate subdivisions are sometimes employed. No scales are shown on the axes of the graph since it is intended as a generic model that indicates internal relativities. During the introduction phase the core product technology is established along with its basic functions. It is rare for a product to address basic functions that are completely novel. For example, it is evident from the example shown in [Figure 17](#) that the models for the early automobiles, or 'horseless carriages' with which they shared many components, were the earlier horse-drawn phaetons and other forms of carriage. The early automobiles inherited the functionality of horse-drawn carriages.



Figure 17 The evolution of complexity: (top) personal transport; (bottom) music reproduction

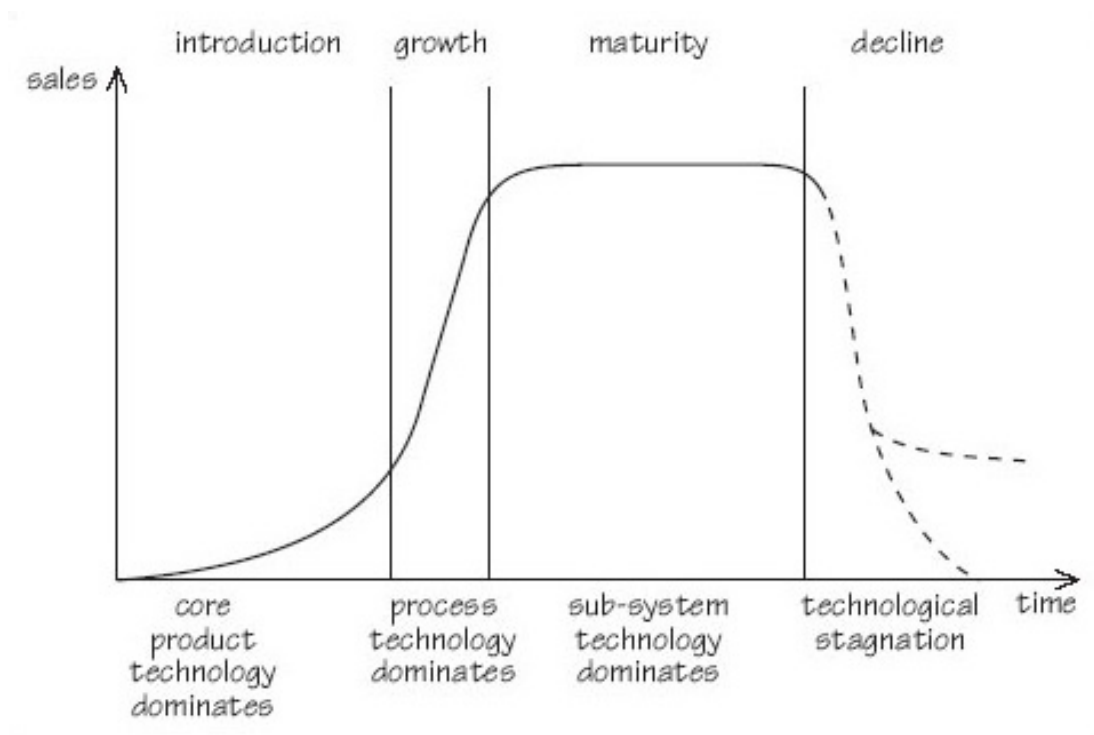


Figure 18 Technology and the product life cycle

From that time until about 1910, the concept of the automobile was elaborated and its performance extended. The industry then entered the rapid growth phase, largely as a result of Henry Ford's application of assembly-line technology to car production (Ford, 1922; Sloan, 1965). By the 1950s the industry had entered the mature phase in the USA and technological innovation was directed at product sub-systems or improvements to the basic production methods. This process has continued up to the present time but, in addition, the development of computing and communications technologies has affected both cars and the way in which they are made. The universality of these technologies has meant that they have driven change in almost every industry and have, for example, fundamentally altered music reproduction, first with the introduction of digital compact discs and then with digital music recordings available for download from internet sites.

The application of computing and communications technologies to both automobiles and music reproduction illustrates a general tendency of systems to become more connected and, as a consequence, for the degree of coupling to increase. A car being driven is connected to traffic monitoring, police surveillance, and possibly global positioning systems, in ways that would have been impossible ten years ago. Similarly, the MP3 music reproduction system works by downloading into its memory music from a personal computer that is likely, in turn, to have been obtained from a website. Previously, cars and music systems were connected to their environment by the actions of humans. These interposed an intelligent buffer between the various elements in what becomes progressively a 'system of systems'. One effect of this is to make the operation of any one system less complex since human actions filter out variation. However, human actions are themselves also a source of variation and complexity which make the operation of the system of systems more uncertain. The replacement of human by technological links increases the size of the system and reduces human interaction. As a result, people are relegated to a position on the periphery of increasingly massive systems over which they can exert less and less control.

1.9 Increasing complication, complexity and risk: summary

The three levels of change problem, simplicity, complication and complexity, can be associated with craft, engineering and systems engineering knowledge. The three categories of change problem represent different levels of uncertainty of what needs to be done and how to do it. The greater uncertainty brings increased risk. Although we tend to be risk averse we will take on greater risk if the returns are commensurate with doing so. Human experience can be divided into three worlds. The physical world of direct experience obeys Newtonian laws of cause and effect. The ways in which this world operates is a 'known problem'. Its operation may be complicated but good explanatory models exist to help predict outcomes. *Complexity* may be added to the complication of the physical world by the actions or interactions of sentient creatures, especially human beings. There do not, as yet, exist good models for the operation of the sub-physical world. This may be a result of it being inherently unknowable or due to 'hidden variables'. Whatever the reason, the behaviour of the sub-physical world remains mysterious and complex. The operation of the supra-physical world of systems, and increasingly of interconnected systems of systems, is similarly mysterious and becoming more so as interconnectedness and, therefore, complexity grow.

SAQ 6

Summarise the reasons why systems engineering is important.

.....

Answer

There are three justifications for systems engineering. The first is to prevent failure. The second justification arose from the need to be able to manage the increasing complexity of systems. The third reason for adopting systems engineering is the fact that customers for systems and stakeholders in them are becoming more unforgiving.

2 What is engineering?

2.1 The development of engineering

Engineering is one important component of systems engineering. In this topic I will examine the development of engineering before presenting a modern view of the subject. Section 3 will then pick up and discuss the idea of systems engineering.

William Shipley, a drawing master from Northampton, was instrumental in founding 'the Society Instituted at London for the promotion of Arts, Manufactures and Commerce' in 1754. This later became the Royal Society for the encouragement of Arts, Manufactures and Commerce and moved to splendid Adams brothers-designed quarters at the Adelphi in London's Strand. The society was to become the inspiration for a number of intellectual bodies based in Britain's fast-growing provincial towns and cities.

Foremost amongst these was the Lunar Society of Birmingham, a group established by 14 prominent local businessmen and others interested in discussing science and technology. Of this society Schofield (1963) states:

More than any other single group, the Lunar Society of Birmingham represented the forces of changes in late eighteenth century England, for the Lunar Society was a brilliant microcosm of that scattered community of provincial manufacturers and professional men who found England a rural society with an agricultural economy and left it urban and industrial. [...] Together they comprised a clearing-house for the ideas which transformed their country materially, socially, and culturally within a generation. They were men of broad interests and their discussions ranged widely, but their major mutual interest was the sciences, pure and applied – particularly as applied to the problems of industry.

This characteristic of the group – its intellectual eclecticism – was typical of the pioneers of the industrial revolution and persisted well into the middle of the nineteenth century. For example, Isambard Kingdom Brunei was responsible for the design of bridges, the Great Western and numerous other railways, locomotives and rolling stock, stations, and the steamships the SS *Great Eastern* and *Great Britain* (Rolt, 1970).

However, the development of scientific and technological knowledge during the nineteenth century meant that, increasingly, it became impossible for one person realistically to pursue an interest in all subjects. This trend was accentuated by the formation of specialist institutes for each discipline, which provided an increasing array of pigeonholes into which engineers could fit themselves. Of particular significance was the separation of science from engineering and technology. The way that this bifurcation occurred is described in Box 4.

Box 4 The separation of science and technology

It was largely due to the reforming zeal of revolutionary and Napoleonic France that the sciences were first organized into their present, tolerably coherent, disciplines. But there is no question that during the nineteenth century the German universities acquired – and deserved – enormous prestige as the world's leading schools for science teaching and research. The ideal of Wilhelm von Humboldt (1767–1835), perhaps the main architect of the success of the German university system in the nineteenth century, was that a university should advance pure learning and that students should acquire a love of disinterested learning – research – by carrying it out for themselves under a master who was an acknowledged scholar. Practical or

vocational studies, that were suggested to require no more than the memorizing of facts, must be excluded. This was an educational creed that went back to Plato and Aristotle and was congenial to the governing elements of all European nations. But von Humboldt's ideal could never be fully realized. As the century wore on, specialization and the educational requirements – the need for more and more school teachers, lawyers, doctors, civil servants and administrators – of a rapidly developing nation state entailed that the Humboldtian programme was progressively diluted. Nevertheless the university ideal of disinterested learning remained and is strongly upheld today. In this was the notion of 'pure science' born. But in practice the 'pure science' was defined administratively; it was the science pursued in universities and not in technical colleges. The model of pure science was imported into America, Britain and other countries by the many students who, having studied at German universities, returned home understandably enthusiastic about German science, research and education. The German technical colleges (*Technische Hochschulen*) and later technical universities could emphasize the importance of free research but they could hardly stress 'pure' learning. They, almost certainly, had far less influence on foreign opinion.

This is not, in any way, a criticism of the admirable system of higher education in Germany. The point is this: a large part of the history and philosophy of science, at least until recently, has been formed in the height of German university practice, a practice substantially followed in the rest of the civilized world. In other words, the history of science reveals the effects of external bureaucratic agencies. To some extent, then, the exclusion of technology from the history of science is a consequence of the exclusion of technology from the German universities.

Source: Cardwell (1994, pp. 8–9)

One result of the process described in Box 4 has been that scientists and philosophers of science have been much more active than technologists and engineers, and philosophers of technology. Philosophers of science, from Francis Bacon (1561–1626) and René Descartes (1596–1650) onwards, have always given prominence to the justification of method, and the claims of various approaches to the accumulation of scientific knowledge have been fiercely debated. In contrast, discussion of engineering method has been a relatively recent phenomenon, having arisen largely in the twentieth century and within the context of systems engineering. Philosophers concerned with technology have been content to examine the broad sweep of development rather than examining methodological development at the level of the project (Basalla, 1988), have examined technology within a social context (Kranzberg and Davenport, 1972) or have adopted an overtly polemical approach (Winner, 1977) in a tradition going back to Mary Shelley's *Frankenstein* (Shelley, 1992 [1818]).

Definitions of engineering are often revealing. Thus R.E. Doherty, President of the Carnegie Institute of Technology (now Carnegie Mellon University) stated: 'Engineering is the art, based primarily upon training in mathematical and physical sciences, of utilizing economically the forces and materials of nature for the benefit of man' (Rae, 1960).

Question 3

What does Doherty's definition of engineering reveal about its status?

Answer

This is a definition that does engineering no favours since it is couched almost entirely in terms of other subjects. First, engineering is not a subject in its own right but is an 'art'. Second, training (notice, not education) for becoming an engineer is mathematics and the physical sciences. Thus engineering is relegated to the status of being about the application of mathematics and the physical sciences to the benefit of humans.

Doherty's definition is typical of many 'conventional definitions of engineering that suggest that it is the application of scientific principles to the optimal conversion of natural resources into products and systems for the benefit of mankind' (Sage, 2000). There are two major problems with definitions of this type. The first is that engineering defined as 'the application of science' fails to recognise that in many instances a technology has been developed and implemented without any scientific foundation. Second, there are four major resource areas or sources of capital that need to be considered:

- natural resources, or natural capital
- human resources, or human capital
- financial resources, or financial capital
- information and knowledge resources, or information and knowledge capital.

2.2 A modern view

Modern attempts to define engineering recognise the importance of the resources identified by Sage, and that the subject can be divided into two components: engineering knowledge – the 'know-what', and engineering process – the 'know-how'. Engineering knowledge is:

[...] the growing body of facts, experience and skills in science, engineering and technology disciplines; coupled to an understanding of the fields of application. [...] It is mainly 'experience-based' knowledge, which is more difficult to describe and communicate than 'codified knowledge' because it must be put into the context of an application.

Engineering knowledge ranges from the more traditional such as civil, mechanical, electrical, chemical, automotive, aeronautical, to the newer such as electronic, communications, medical, bio-technical. These subjects are being added to regularly.

(Royal Academy of Engineering, 2000)

Engineering know-how is seen in terms of problem solving in an output standard for engineering graduates published by the UK Engineering Professors' Council (EPC, 2000). This standard is:

based on the generic procedures carried out by an engineer in solving an engineering problem and delivering the solution. Engineering problem solving is an iterative task involving creativity and the application of knowledge and understanding. Broadly, an engineer needs to be able to identify and describe the problem

that is to be solved ... The solution will have a specification with parameters that require evaluation, a process that relies on the engineering skills of conceptualisation, determinable modelling and analytical representation. Delivery of the specified solution draws on other skills including the verification of conceptual models by experimentation with physical models.

The standard is structured around four types of model and their transformations.

Conceptual models describe needs, ideas and the current situation. They are usually pictorial: flowcharts, schematics, state transition diagrams, decision trees, etc.

Determinable models help to evaluate solutions. 'Determinable' is an unfortunate choice of word since it means 'capable of being definitely ascertained' (OED, 1989), whereas some useful engineering models are stochastic, i.e. based on statistical probability. The intended sense is 'calculable'. Calculable models include finite element analyses, simulations, and many analyses of key 'figures of merit' such as costs and benefits, risk, reliability, maintainability, service levels, capacity, performance, etc.

Physical models help to investigate critical aspects of solutions. They include prototypes and proofs of concept, scale models and mock-ups, 'breadboards', etc.

Specification models define a solution precisely. They take forms such as formalised languages and bill-of-materials databases. The first three types of model may also specify; for example, conceptual models are often used to specify information systems requirements. This is only possible when a model has a formal structure (a meta-model) whose elements can be supported by precise definitions. Free-format models can describe but not specify; you should watch out for sketches masquerading as specifications. Without the definitions, it is not possible to verify the model, for example by comparing it with samples of real information and by checking that each system output can be derived from it. A secondary function of the definitions is to communicate any concepts for which the diagram notation is not rich enough.

In the same year a joint Royal Academy of Engineering/Engineering Council Working Group clearly defined the engineering process as 'the creative process which applied knowledge and experience to seek one or more technical solutions to meet a requirement, solve a problem, then exercise informed judgement to implement the one that best meets constraints' (Royal Academy of Engineering, 2000, p. 32). The Working Group summarised the engineering process with the diagram shown in [Figure 19](#).

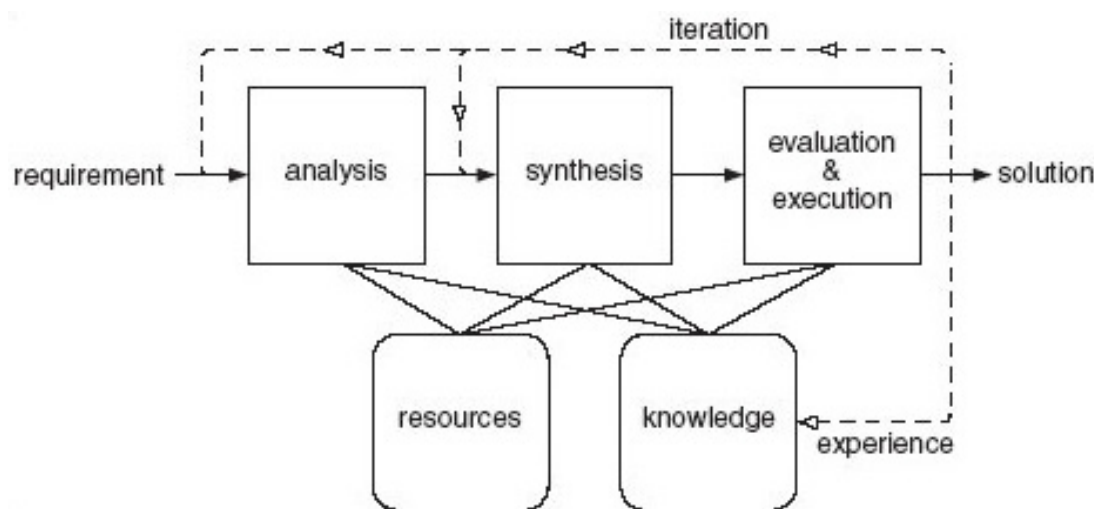


Figure 19 The engineering process. Source: Royal Academy of Engineering (2000)

Engineering as a process can also be viewed from a 'design' perspective. For example, Pugh (1991) sets his methodology, shown in [Figure 20](#), within the context of a design problem. He makes an important distinction between traditional engineering which leads to practical design and his concept of 'total design'.

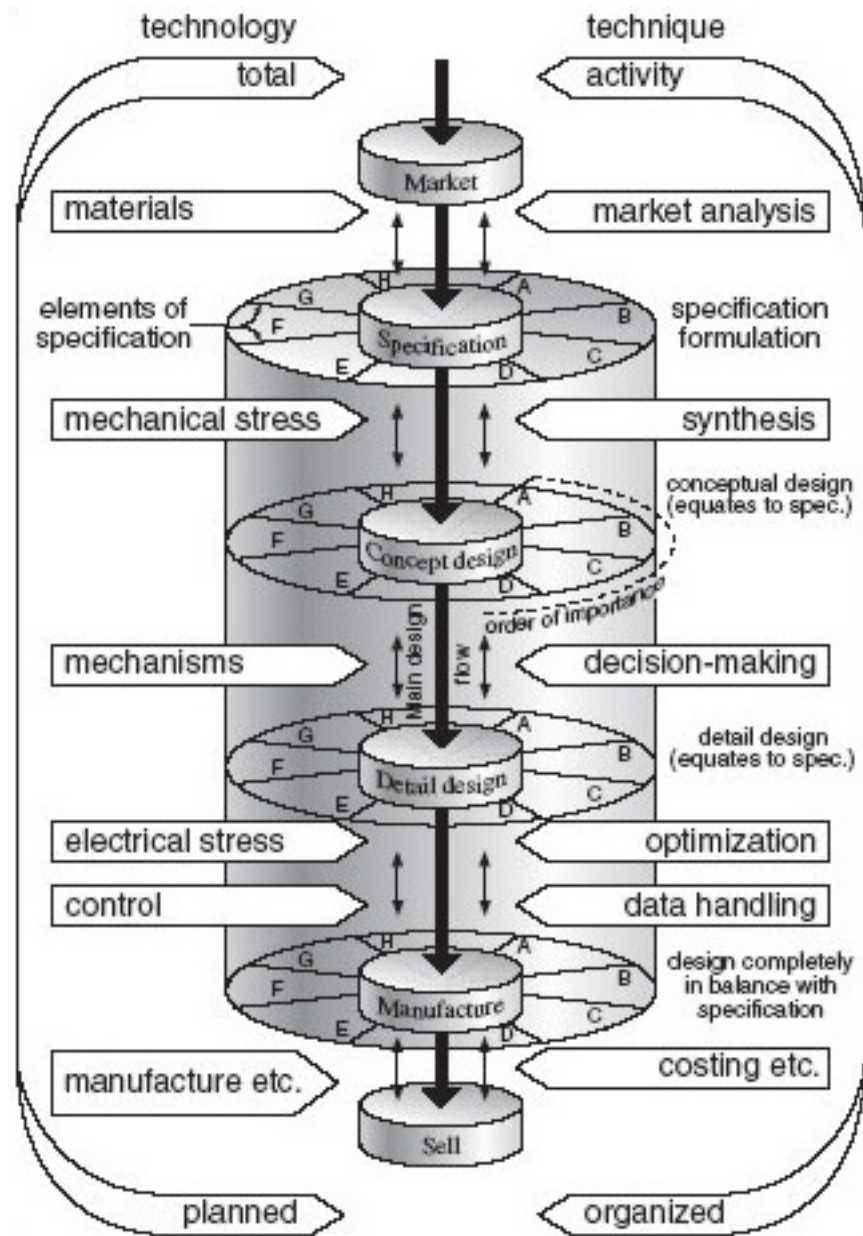


Figure 20 Pugh's total design activity model. Source: Pugh (1991)

One way to consider the characteristics of a phenomenon is to contrast it with another, closely related phenomenon. The distinctions between engineering and scientific process are set out in Box 5.

Box 5 The differences between engineering and scientific processes

Engineering process	Scientific process
Invention, design, production	Discovery (mainly by controlled experimentation)
Analysis and synthesis of designs	Analysis, generalisation and synthesis of hypothesis
Holism, involving the integration of many competing demands, theories, data and ideas	Reductionism, involving the isolation and definition of distinct concepts
Activities always value-laden	Making more-or-less value-free statements
The search for, and theorizing about, processes (e.g. control, information, networking)	The search for, and theorizing about, causes (e.g. gravity, electromagnetism)
Pursuit of sufficient accuracy in modelling to achieve success	Pursuit of accuracy in modelling
Reaching good decisions based on incomplete data and approximate models	Drawing correct conclusions based on good theories and accurate data
Design, construction, test, planning, quality assurance, problem-solving, decision-making, interpersonal, communication skills	Experimental and logical skills
Trying to ensure, by subsequent action, that even poor decisions turn out to be successful	Using predictions that turn out to be incorrect to falsify or improve the theories on which they were based

Source: Royal Academy of Engineering (2000, p. 34)

SAQ 7

Examine critically the two lists of key processes in Box 5. List your comments on them. How would you summarise the difference between engineering and science?

Answer

There seem to me to be two levels of comment that can be made about the engineering process list. The first level concerns the structure of the list as a whole.

1. Not all of the items on the list are processes. Neither 'Holism' nor 'Activities always value-laden' are processes. They would, perhaps, be better described as attributes.
2. The items on the list seem to be uneven in level. For example, the list includes 'design', the 'analysis and synthesis of designs' and then 'design, construction'.
3. As '2' above suggests, there is some repetition of items.

4. The list lacks structure – or at least one that is discernible.
5. Some of the statements are disputable. For example, what is meant by the statement ‘Activities always value-laden’? Why in the last item on the list do engineers only ‘try’ to achieve an outcome?

As far as the scientific processes list is concerned:

1. Certainly science tries to make discoveries, but a better way of putting this aim would be to state that it ‘seeks to add to knowledge’.
2. Science is concerned with hypotheses and research problems and its progress is driven by formulating and testing them.
3. Reductionism does not just concern ‘concepts’.
4. Science is not ‘value-free’.
5. Experimental and logical skills, though necessary, are not a process.

I would consider that the aim of science is ‘understanding’ and to advance knowledge whereas engineering is purposeful, seeking to produce effective solutions to real-world problems.

2.3 Summary and conclusions

This topic has addressed the question ‘What is modern engineering?’ The conclusion must be drawn that, until recently, engineers were content with fairly simplistic definitions of their profession, thinking that it consisted of little other than craft skills or practical experience grafted on to a knowledge of mathematics and appropriate natural sciences. It has been methodologically naive, and definitions of the processes of engineering either lack detail ([Figure 19](#)) or have been constructed in terms of ‘design’ ([Figure 20](#)).

By contrast, ‘systems engineering’ has a plethora of methodologies – perhaps too many for its own good – and in Section 4 I will examine the development of systems engineering and its methodologies. The next section examines the second component of systems engineering by answering the question ‘What is systems?’

3 What is systems?

3.1 Introduction

As you would expect, since this course deals with *systems* engineering, it embodies the principles and methods associated with a systems perspective. So it is important that you understand systems and the systems perspective at the beginning of the course.

To have engineered a system successfully, all its features – the technology, control systems, people and related aspects of the physical environment – have to contribute to the achievement of its objectives. In other words, it has to operate in an integrated, coherent way. It has to meet the requirements of all stakeholders. There are also things that it must *not* do. For example, it should not exceed its agreed or projected running costs; it should not, in achieving its requirements, wantonly harm individuals or the physical environment, and so on. These sets of requirements and constraints can be represented by the three models shown in [Figure 21](#).

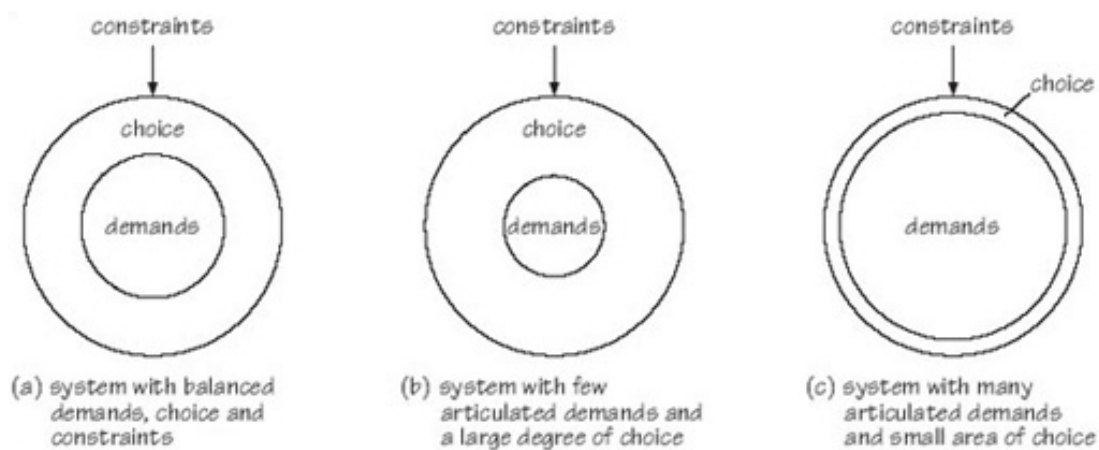


Figure 21 Patterns of demand, choice and constraint in systems engineering

SAQ 8

The demands, choices and constraints associated with a system can arise from any of the stakeholders in that system. If stakeholders are defined as an identifiable group of people or an organisation, or organisations, having a legitimate interest in the process or outcomes of a systems engineering project (see also Box 6), identify the stakeholders in the Millennium Bridge project.

Answer

The stakeholders involved in the Millennium Bridge were as follows.

1. Those who put up the money:
 - The Millennium Commission through the Millennium Bridge Trust
 - The Corporation of London through the Bridge House Estates Trust
 - The Hong Kong and Shanghai Banking Corporation (HSBC)
 - The *Financial Times* (which sponsored the original competition)
 - and other contributors of funds.
2. Those involved in the bridge's design or construction:

Ove Arup
Foster and Partners
Sir Anthony Caro
Sir Robert McAlpine
Monberg Thorsen
subcontractors.

3. Local authorities, the Borough of Southwark.
4. Those concerned with the bridge's context: e.g. the Tate Modern's architects, Herzog & de Meuron.
5. The public wishing to use the bridge.
6. River users.

Box 6 stakeholders

Mitroff (1980) defined the stakeholders in a system as all those:

having an interest in the change being considered
who stand to gain from it
who stand to lose from it.

The stakeholders in any system can be classified into one of four categories:

1. Those responsible for its design and development, for example the project managers, system engineers, communications experts, technical authors.
2. Those with a financial interest in the system, responsible for its sale or its purchase, for example the marketing manager, the buyer.
3. Those responsible for its introduction and maintenance within an organization, for example training and user support staff.
4. Those who have an interest in its use, for example user managers and all classes of users: primary (those who are likely to be frequent hands-on users of the system); secondary (those who are occasional users of the system or who make use of it through an intermediary); tertiary (those who are affected by the introduction of the system or who will influence its purchase but who are unlikely to be hands-on users).

Adapted from Macauley (1996)

The constraints define the area in which the system can legitimately operate. The articulated demands or requirements of the system are at the core of the system. They, together with the constraints, define the amount of choice that the systems engineers will enjoy. In practice, and even with the best endeavours of the various stakeholders in the system, demands and constraints are rarely articulated in a way that the clarity of the circles in [Figure 21](#) might suggest. One way of looking at systems engineering is to regard it as a process of progressively refining demands (requirements) and constraints in such a way that design choice is eliminated. This process of progressing from an initial, fuzzy concept of what the system will do to a final, implemented and ambiguity-free system is illustrated in [Figure 22](#).

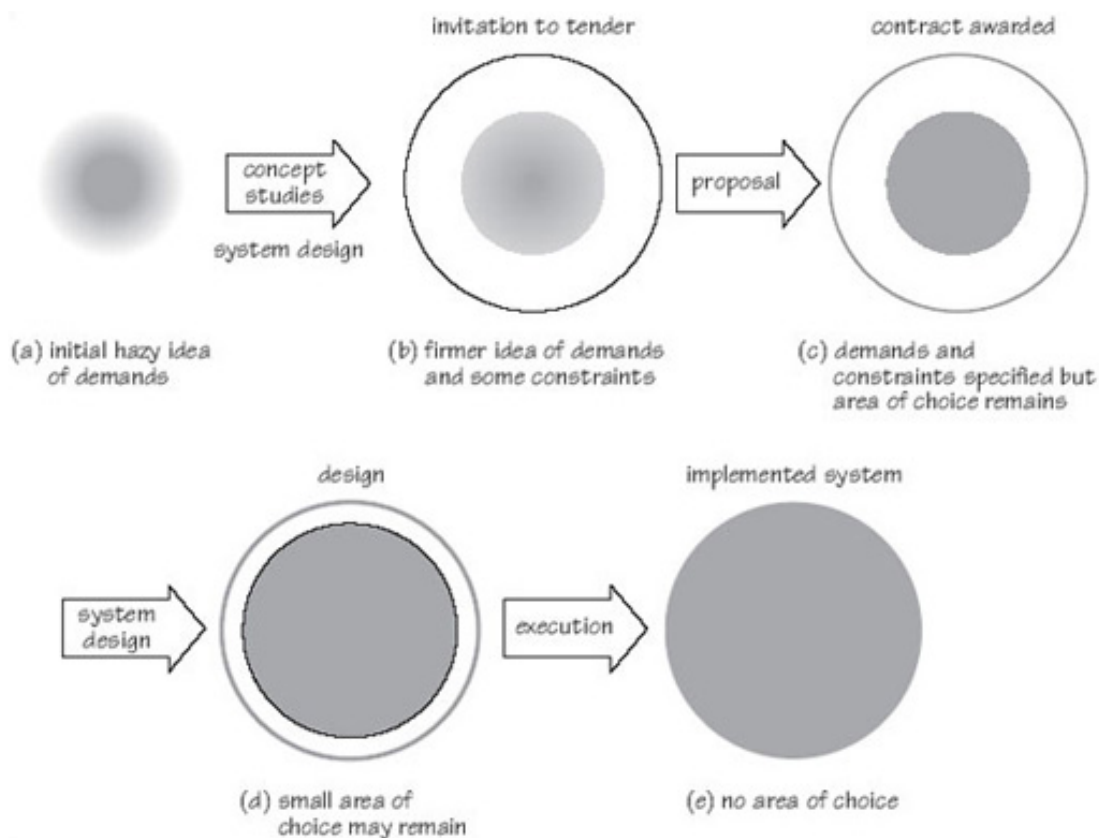


Figure 22 The progressive definition of the system

SAQ 9

Into which of the three categories shown in [Figure 21](#) would you place the Autodesk and Millennium Bridge examples discussed in Section 1?

.....

Answer

I would classify both Workcenter and the Millennium Bridge as type 'b' systems engineering projects.

The success of a systems engineering activity is judged by:

- how well the outputs of the activity fulfil demands and how well the design of the system has avoided violating constraints, and
- how successfully the systems engineers have implemented the choices that were available to them, and
- the quality of the systems engineering process.

Although the quality of the systems engineering process may be regarded as a tertiary measure of a systems engineering project, it is the foundation of the primary and secondary measures of success.

Adopting a systems approach addresses the need to establish and maintain the internal coherence of the project and its external requirements. The systems approach consists of:

- a set of concepts that provide a philosophic basis for the approach
- a methodology that forms a framework for designing, developing, operating the system and managing change
- a set of techniques, used within the context of the methodological framework, that provide a toolkit for planning, analysis and design work.

The relationship of these three elements of the systems approach is illustrated in [Figure 23](#). Each of these elements of the systems approach will be discussed in turn.

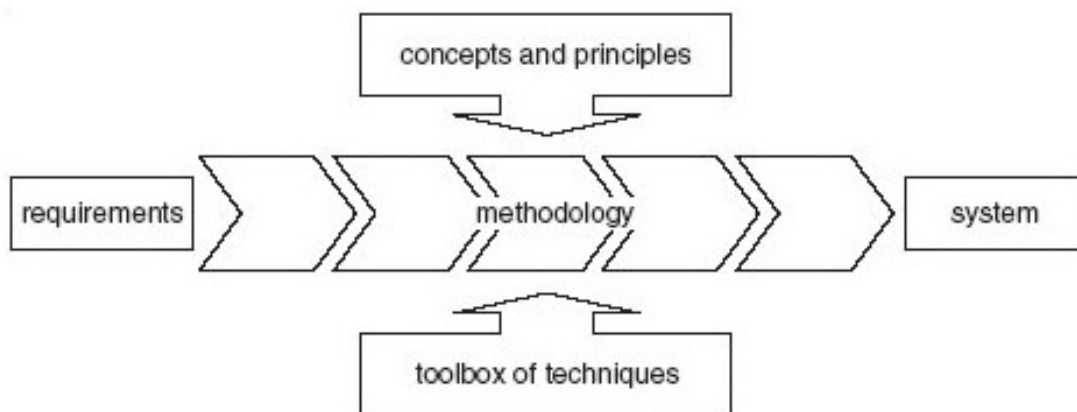


Figure 23 Systems as concept, methodology, technique

3.2 Systems concepts: system

The word 'system' is from the Greek word *συστήμα* meaning a complex, organised whole. It has been used in this sense throughout history, and the *Oxford English Dictionary* records examples of usage dating from the early eighteenth century. [Figure 24](#) shows a simplified diagram of a typical system. It indicates the boundary of the system, which divides those elements that are considered to be within the boundary, and therefore part of the system, from those elements in the environment of the system. I will discuss each of these features of systems in turn.

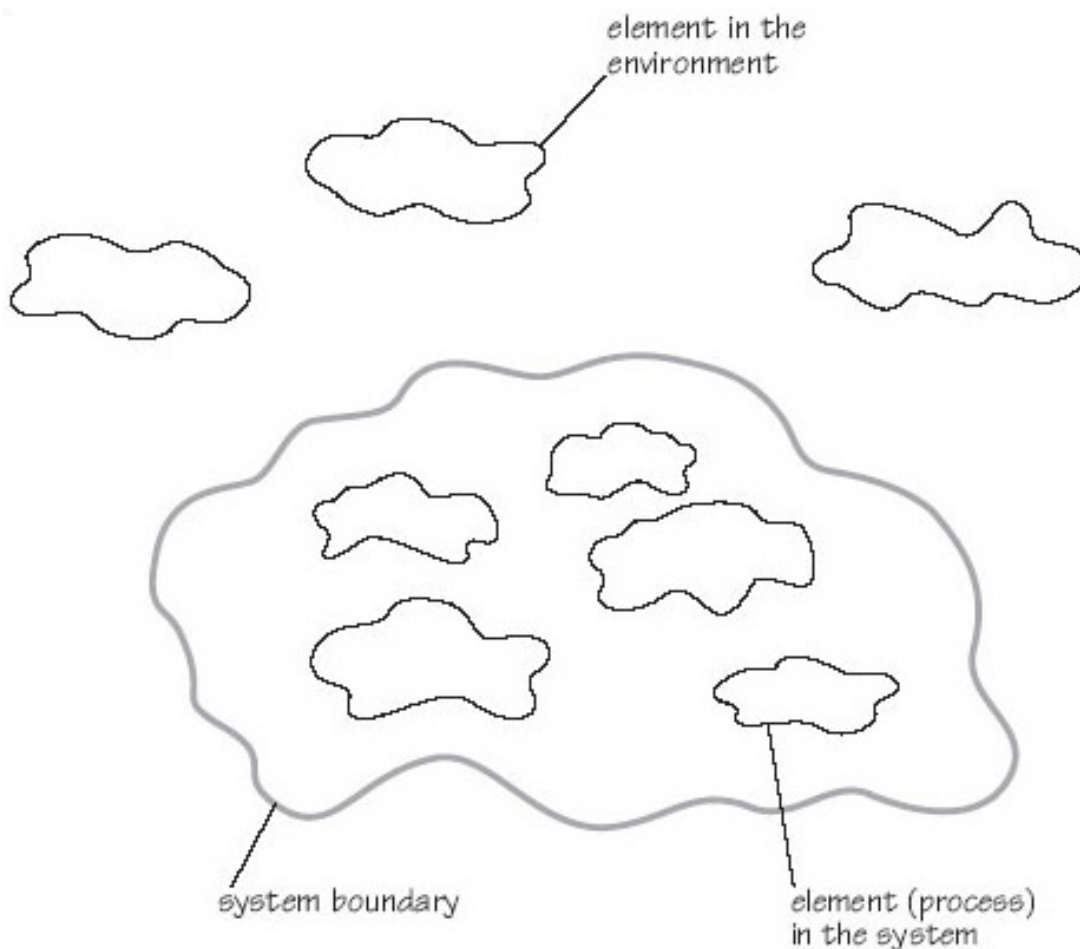


Figure 24 A simplified diagram of a system. Source: BS/ISO/IEC 15288, p. 56

Drawing or positioning a *system boundary* is a difficult matter of judgement involving decisions about what is to be included and what should be placed in the environment. The difficulty of this act of judgement reflects its importance to the success of the system, since the placement or delineation of the boundary reflects the perspective of the system's designer. For example, in the Autodesk Workcenter project the designers within the company drew the 'boundary of the product' around what could be placed in the box (literally). They considered that they were designing a piece of software and, of course, to some extent that was what they were doing. Had they adopted a different, broader perspective they would have included within the product boundary the consultancy needed to achieve its successful implementation. It is an easy matter to redraw the boundary of a system on paper at a very early stage of development. However, as projects progress the boundary becomes embedded in the design concept, investment is made, and it becomes progressively more difficult to alter the position of the boundary. Positioning the system boundary is vitally important in systems engineering and is recognised in BS ISO/IEC 15288 'Systems engineering – system life cycle processes'. This distinguishes:

- the system of interest: the system whose life cycle is of interest to a particular project
- enabling systems – a system that contributes to the system of interest during one or more of the stages of its life cycle but which does not contribute directly to its operation. For example, a production system may be required to realise the system. Enabling systems have their own life cycles and will probably, at one time, have been, themselves, systems of interest. The standard notes that a systems

engineering project may also have to take responsibility for creating an enabling system. (ISO/IEC 15288, p. 56)

- systems in the operating environment: these are systems with which the system of interest interacts when it is operational. The nature of the interaction may be goods and services, energy, or information.

The Standard's view on the relationship of these three types of system is shown in [Figure 25](#).

My perspective is that all systems are designed. This is demonstrably the case for systems that are engineered to be of utility, but is also true of systems that are formulated for research or analytical purposes. In both instances decisions on what is included in the system, and therefore what is in its environment, and the establishment of the relationship between the elements within the boundary constitute design decisions. Therefore, whether explicitly designed or not, systems are in the eye of the beholder.

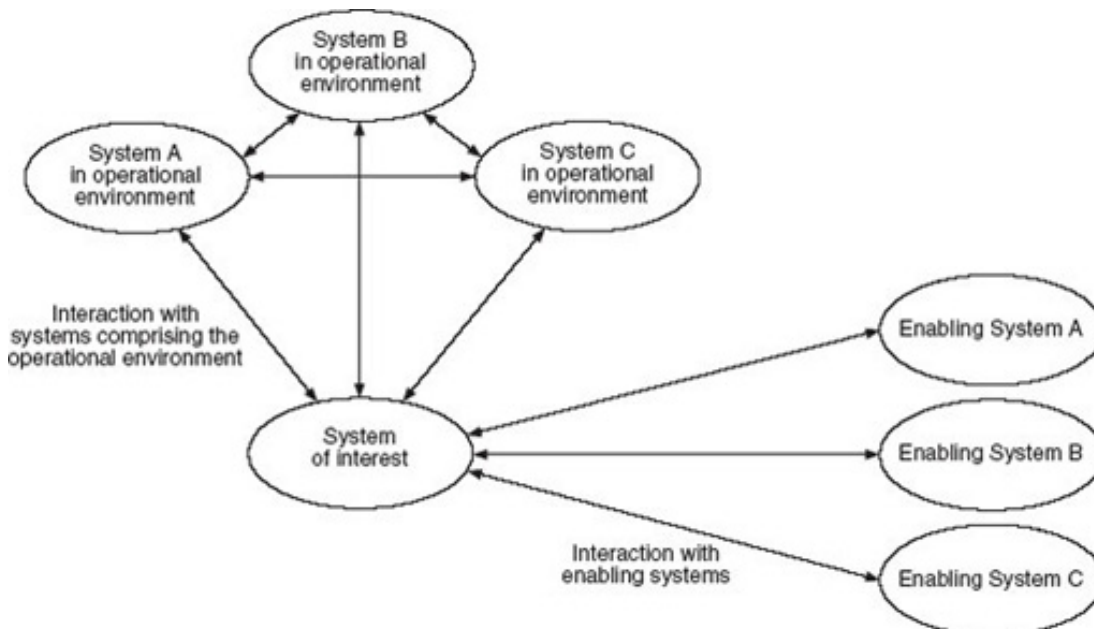


Figure 25 System-of-interest, its operational environment and enabling systems

In 1928/29 the Belgian surrealist painter René Magritte exhibited a work that he called *The Treachery of Images*. It depicted a smoker's pipe with the cautionary label '*Ceci n'est pas une pipe*' (*This is not a pipe*). Preoccupied with the flimsy veil between image and reality, Magritte was making the point that an image of a thing is not the thing itself. Magritte's image is reproduced in [Figure 26](#) with another level of caution!



Figure 26

To go beyond Magritte's image and answer the question 'What, in reality, is a pipe?' it would be necessary to get hold of a real pipe and demonstrate it. Setting aside solipsist objections we might conclude 'OK, that really is a pipe'. As the eighteenth century empirical philosopher John Locke stated, 'External objects furnish the mind with the ideas of sensible qualities ...' (Locke, 1997 [1706]) which are all those different perceptions which they produce in us. We can experience the reality of a pipe by looking at it, touching it and (perhaps especially) by smoking it and smelling it. However, none of our senses, neither singly nor in combination, can be used to detect the reality of a *system*.

While we may accept that knowledge is derived from our senses, nobody sees, feels, hears, tastes or smells a system. What we conceive are objects, people, information, ideas and relationships to which we give the label 'system'. In attaching the label 'system' to them we are expressing explicitly or implicitly:

1. a purpose for the assemblage
2. that a relationship or set of relationships exists between the elements in the assemblage.

A system is therefore an intellectual artefact and the label 'system' can be used for two purposes. First, it is a convenient way of putting a conceptual boundary around a disparate set of elements. For example, 'the public transport system' might include within its boundary railways, air transport, buses and coaches, and taxis. Second, using the label 'system' suggests a (different) way of looking at the world (Boehm, 1988). The systems viewpoint is one that emphasises purpose and process.

The image shown in [Figure 27\(a\)](#) represents a fountain pen, but it might also be described as a system for turning thoughts into marks on paper (b) that another person can (more or less) read. The object label 'fountain pen' emphasises existential form, whereas the system label tells us what it is for and the transformation (thoughts into marks on paper) that it is part of. Using the systems viewpoint is extremely valuable in engineering since it

can prevent premature judgements being made. Setting out on a project with the objective of 'designing a new fountain pen' is a very different proposition from 'designing a system for turning thoughts into marks on paper'.

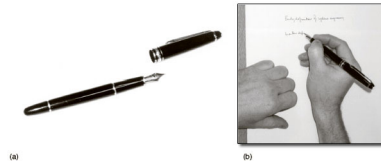


Figure 27 (a) A fountain pen or (b) a system for turning thoughts into marks on paper

Starting from the requirement of a new system to put marks on paper might produce a solution radically different from a fountain pen. Another important attribute of the systems viewpoint is that it encourages a holistic perspective, which I will discuss in Section 3.3. Each of the elements in the system is affected by being part of the system, and the way that the system operates would change if an element were to be removed. Elements of a system can be decomposed into a more detailed set of sub-elements that are included as part of the system. The level of analytical detail will depend on the perspective of the analyst and the purpose that he or she has in mind. The elements that should be included as part of the system's environment are those that influence the system in some important way, but over which the system has no direct control. Each element in a system can be regarded as a process, and the system itself consists of processes that are coupled together in an identifiable way. By process, I mean a set of activities that produce an identifiable output. A process may be formally defined, and in systems engineering most processes will be of this type. One of the objectives of systems engineering methodology is to make processes and their associated outputs explicit. This characteristic will be examined in more detail later.

All the elements in a system and its environment are connected in some way. The connection may be physical – raw materials, components, assemblies and products move from one element to another. But the connection or relationship can take many forms – organisational power and authority, information and political influence.

The nature and form of the relationship between the processes in a system express the intent of its designers, whether the system is utilitarian or analytical in character.

Therefore, as I suggested earlier, as its creator and the identifier of the system, the engineer, researcher or analyst is an important feature of the system. This is not to imply that the whole thing is completely subjective; that would be true to an extent neither greater nor less than any other form of engineering or analysis. But it does mean that the perspective and purpose of the system's creator have to be taken into account, as [Figure 27](#) suggests.

To summarise, a system may be formally defined as:

1. an intellectual artefact that consists of
2. an assembly of processes that are
3. connected in a determinable relationship, which
4. has a purpose that is
5. either utilitarian or analytical in character.

This definition is wider than that proposed in the standard – 'a combination of interacting elements organized to achieve one or more stated purposes' (ISO/IEC 15288, p. 4) – but has recognisably common characteristics. The main differences are that the T837_1

definition explicitly reflects the subjective character and recognises that systems thinking can be used for analytical as well as design purposes.

3.3 System concepts: holism

One of the distinguishing features of the systems approach is its attempt to be holistic – to include all the elements in the picture at each level at which the system operates. The premature exclusion of important elements can be dangerous and can lead to, for example:

- a purchasing manager being so keen to drive down raw material and component costs that he or she causes quality and production problems in construction of the system
- engineers thinking that the design of the control system can be left until later, causing expensive hardware to stand idle waiting for software
- measures of performance creating an excess inventory and so reducing the cash flow of the system as a whole.

In each case, a proper regard for the wider system – taking a holistic view – would have avoided a sub-optimal course of action. The way that different functional specialists might view the same systems engineering problem is shown in [Figure 28](#).

The systems approach is often contrasted with what is frequently called the 'reductionist' approach of traditional science. This characterises science as being concerned with reducing problems to their smallest elements and in doing so to ignore synergistic effects that occur at each level of a deconstruction.

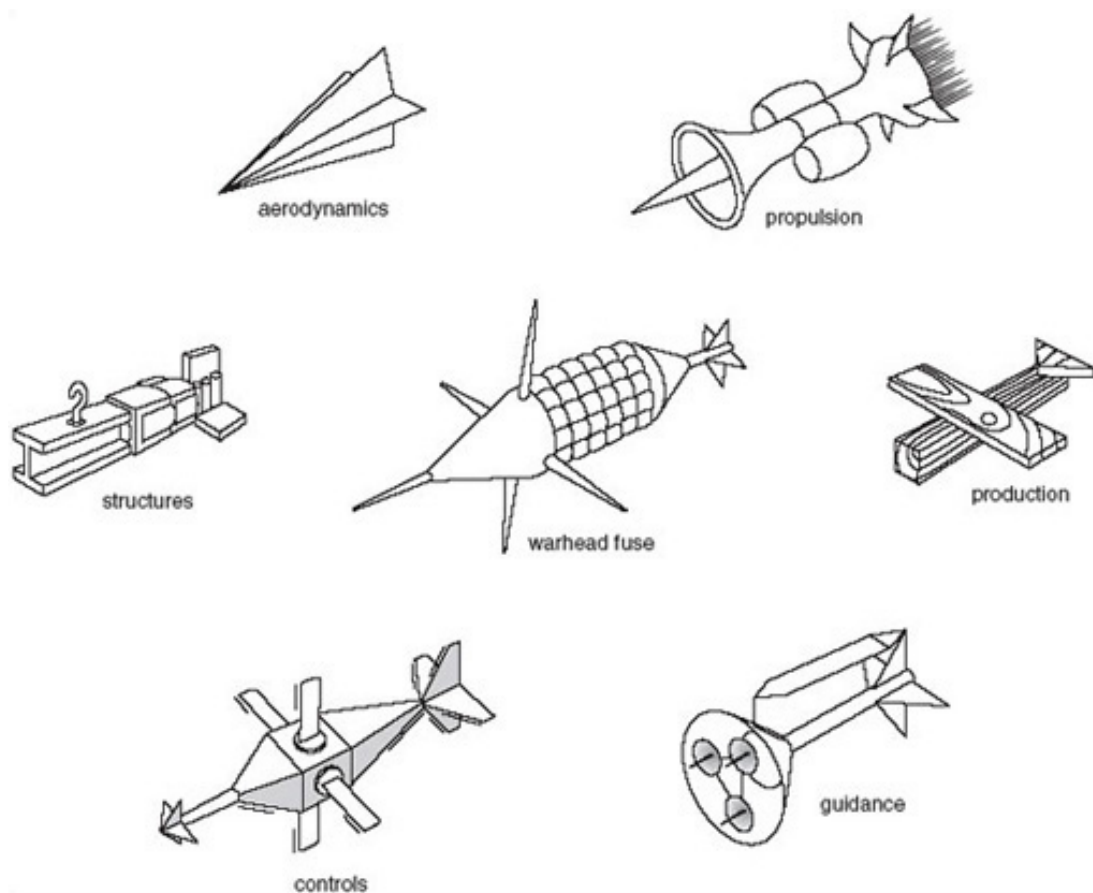


Figure 28 How different functional specialists view the same systems engineering problem

Checkland (1993, pp. 44–58) in particular traces the origins of reductionism back to Descartes' *Second Discourse*. In this, Descartes states four rules for developing knowledge:

The first was never to accept anything as true if I had not evident knowledge of its being so; that is, carefully to avoid precipitancy and prejudice, and to embrace in my judgement only what presented itself to my mind so clearly and distinctly that I had no occasion to doubt it.

The second, to divide each problem I examined into as many parts as was feasible, and as was requisite for its better solution.

The third, to direct my thoughts in an orderly way; beginning with the simplest objects, those most apt to be known, and ascending little by little, in steps as it were, to the knowledge of the most complex; and establishing an order in thought even when the objects had no natural priority one to another.

And the last, to make throughout such complete enumerations and such general surveys that I might be sure of leaving nothing out.

(Anscombe and Geach, 1970, pp. 20–21)

This extract suggests that to draw a firm distinction between supposedly reductionist science and a holistic systems approach is too simplistic.

It is the second of Descartes' rules that is said to sow the seeds of reductionism and the development subsequently of scientific method based on it. However, the characterisation

of Descartes' thinking as reductionist does not accord with the facts. In *Rules for the Direction of the Mind*, which was written before 1630, and first published in the *Opera Posthuma* in 1701, Descartes clearly proposes that we should seek to understand what he terms 'simple natures', but importantly also that:

... the conjunction of these simple natures with one another is either necessary or contingent. It is necessary when one is implicitly contained in the concept of the other, so that we cannot distinctly conceive of either if we judge that they are separated [...] A combination of natures is contingent when they are not conjoined by any inseparable relation; as when we say that a body is animated, a man is clothed, etc.

(Anscombe and Geach, 1970, p. 174)

Descartes is making the point that it is important not only to give attention to individual elements but also to consider how these elements fit together. This is an entirely appropriate position for systems engineering. The whole (system) is understandable only in terms of its components and the proper relationship between them, and the components of a system may have no meaning other than as part of the system.

The desire to separate and classify rather than to regard the natural world as a continuum began in the sixteenth century, gathered momentum during the eighteenth, and reached its apogee in the nineteenth (Foucault, 1974, pp. 125–165). Its origins are to be found in the work of Descartes, Bacon and biologists such as Linnaeus, but the urge to classify quickly spread to related human activities such as the disciplines of science themselves. A classification of the sciences first proposed by the influential Victorian scientist William Whewell in 1840 is shown in [Figure 29](#). In a similar manner, the work of business organisations is routinely divided into functional specialisms such as product development, marketing, operations, and finance. This arrangement, has begun to alter slightly over the past 15 years as the barriers to effective customer focused business that functions represent has been recognised. However, in many, if not most businesses traditional forms of organisation remain dominant.

CLASSIFICATION OF SCIENCES		
Fundamental Ideals or Conceptions	Sciences	Classification
Space.....	Geometry.....	Pure Mathematical Sciences
Time.....		
Number.....	Arithmetic.....	
Sign.....	Algebra.....	
Limit.....	Differentials.....	Pure Motional Sciences
Motion.....	Pure Mechanism.....	
	Formal Astronomy.....	
Cause		
Force.....	Statics.....	Mechanical Sciences
Matter.....	Dynamics.....	
Inertia.....	Hydrostatics.....	
Fluid Pressure.....	Hydrodynamics.....	
	Physical Astronomy.....	
Outness		
Medium of Sensation.....	Acoustics.....	Secondary Mechanical Sciences (Physics)
Intensity of Qualities.....	Formal Optics.....	
Scales of Qualities.....	Physical Optics.....	
	Thermotics.....	
	Atmology.....	Analytico-Mechanical Sciences (Physics)
Polarity.....	Electricity.....	
	Magnetism.....	
	Galvanism.....	
Element (Composition)...		
Chemical Affinity.....		
Substance (Atoms).....	Chemistry.....	Analytical Science.
Symmetry.....	Crystallography.....	Analytico-Classificatory Sciences
Likeness.....	Systematic Mineralogy.....	
Degrees of Likeness.....	Systematic Botony.....	Classificatory Sciences
	Systematic Zoology.....	
Natural Affinity.....	Comparative Anatomy....	
(Vital Powers)		
Assimilation		
Irritability		
(Organization).....	Biology.....	Organical Sciences
Final Cause		
Instinct		
Emotion.....	Psychology	(Metaphysics)
Thought		
Historical Causation....	Geology.....	Palætiological Sciences
	Distribution of Plants and Animals.....	
	Glossology.....	
	Ethnography.....	
First Cause.....	Natural Theology.	

Figure 29 Whewell's classification of sciences. Source: Whewell (1968 [1858], p. 190)

The logic of such functional specialisation is that it is more efficient. A person, a department, or an organisation that focuses on one area of activity will tend to perform better than others that are 'jacks of all trades and masters of none'. As Adam Smith famously put it, 'The division of labour, [however], so far as it can be introduced, occasions, in every art, a proportional increase in the productive powers of labour.' (Smith, 1976 [1776], p. 9). The systems approach is based upon the alternative and equally important logic that any given circumstance, whether academic or in business, benefits from multidisciplinary, and when the task is to develop an integrated system that will operate in a closely coupled manner the adoption of a holistic approach is essential. Holism is of particular importance to systems engineers, as taking a holistic view of a system that is proposed or being developed will help to reduce the potential problems associated with incomplete knowledge, risk and emergence that have been discussed earlier.

3.4 Systems concepts: structure

As suggested earlier, the structure of a system is its functional or physical arrangement; the term that is often used in systems engineering is 'architecture'. The architecture of a system can be deconstructed to reveal its constituent elements. I suggested in Section 1 that an existing knowledge base has an important bearing on the way in which a change problem is perceived. The way that this is conceived by one armaments system integrator is illustrated in [Figure 30](#). *Process* is connected with the dynamic behaviour of the system. *Structure* is those processes that change relatively slowly; dynamic behaviour changes relatively quickly.

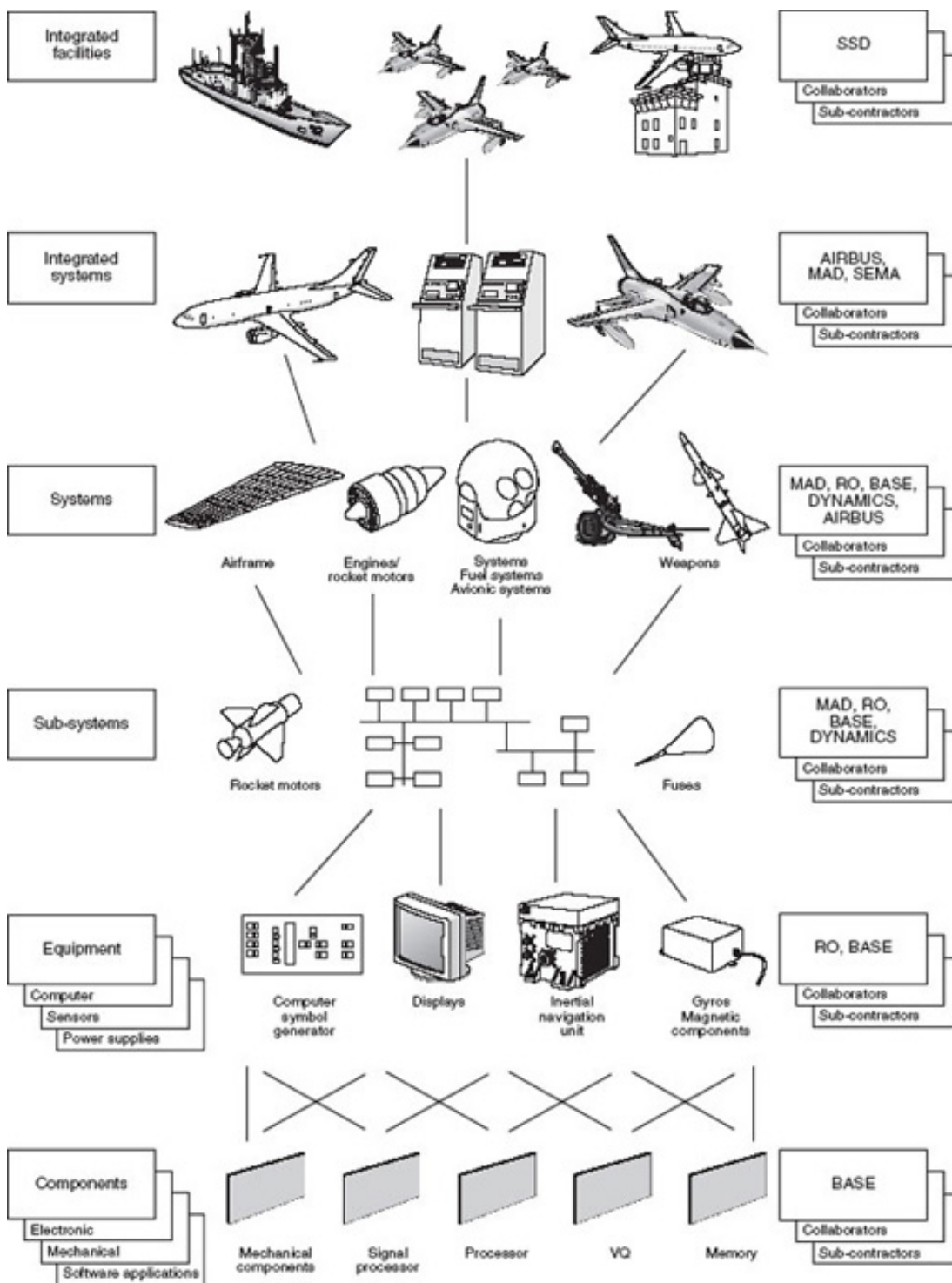


Figure 30 An example of system structure hierarchy. Adapted from Royal Ordnance (1997)

The ISO/IEC 15288 standard adopts a three-layer approach to the decomposition of a system of interest. The top layer is the system of interest itself; this is composed of many systems which, in turn, are made up of many system elements. A system element is '... a discrete part of a system that can be implemented to fulfil specified requirements.' (ISO/IEC 15288, p. 4).

Both the T837 and the ISO/IEC 15288 definitions of systems and system structure can be stated formally (Kaposi and Myers, 1994, p. 15). A system **S** can be represented as

$$\mathbf{S} = (\mathbf{E}, \mathbf{R_E})$$

where

$\mathbf{E} = \{e_1, e_2, \dots, e_n\}$ is a set of elements

$\mathbf{R_E} = \{r_{E1}, r_{E2}, \dots, r_{Em}\}$ is a set of relationships on the elements

The arrangement of the elements in a system is usually represented by simple hierarchical diagrams which simply imply a 'contains' or 'includes' relationship. In fact, the structure of a system can equally well be considered in terms of its dynamic behaviour.

3.5 Systems concepts: dynamic behaviour: input-transformation-output

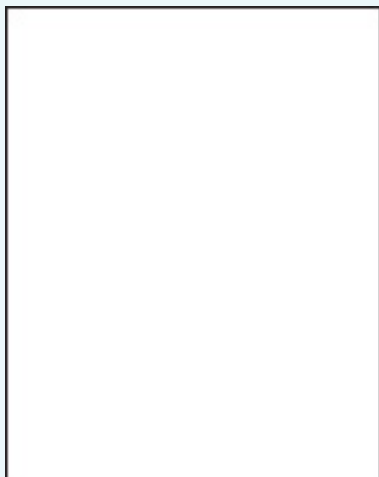
Utilitarian systems, as previously discussed, are the means we use to transform resource inputs into useful goods and services. Any system can be divided into a set of input-transformation-output blocks. These are usually represented as in [Figure 31](#). This way of looking at systems can be used as an analytical and design tool.



Figure 31 Input-process-output

SAQ 10

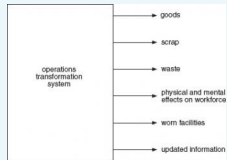
Take a plain blank sheet of paper. In the middle draw a rectangle like this:



Label your rectangle 'operations transformation system'. From your knowledge of operations, make a list down the right-hand side of the sheet of all the outputs of an operations system that you can think of. Join these to your process rectangle with arrows coming from the rectangle.

Answer

My outputs were:



Transformation processes should always have a verb attributed to them: forming, painting, assembling, packaging are all examples of transformation processes. Transporting is another example, since a component or product is 'transformed' from being in one place to being in another.

Outputs are what the system is all about – or at least some of them are. Unless a utilitarian system produces goods, services and information, and produces them in the right quantity, at the right time, of the appropriate quality and at planned cost, its purchasers or creators might as well save their time, money and effort.

SAQ 11

(a) From your knowledge of operations, make a list down the left-hand side of the diagram that you drew in answer to SAQ 10 of all the important inputs to the system. Connect them to the operations system rectangle with arrows.

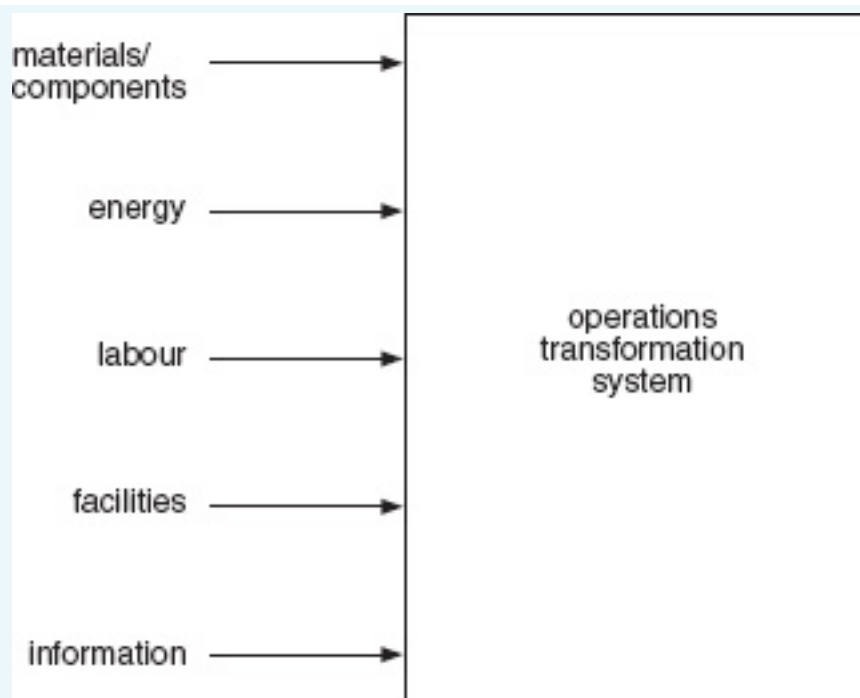
(b) The inputs to the transformation system are of three sorts:

- those that it uses;
- those that it transforms;
- those that it uses for planning and control.

Decide which of the inputs in your answer to (a) are used by the system, which are transformed, and which are used for planning and control. Then compare your diagram and responses to the ones given in the answers to SAQs 10 and 11.

Answer

(a) My diagram looked like this:



(b) The inputs that the system uses are:

energy
labour
facilities.

The inputs that the system transforms are:

materials/components.

The input that the system uses for planning and control is:

information.

There are three reasons for getting you to carry out the input–transformation process–output exercise in SAQs 10 and 11.

- to make you think about systems engineering in terms of inputs, processes and outputs;
- to introduce input–output diagramming, which is a common technique used in the analysis and development of systems;
- to illustrate how using tools of this kind can help you to think in a holistic way about a system and can promote further analysis.

Look again at the outputs shown in the diagram associated with the answers to SAQs 10 and 11. Some of the outputs are planned, some are accepted and some are undesirable. It would be possible, for example, to carry out a further, more detailed, analysis on the scrap and waste producing sub-system.

Input–output diagrams are good for identifying and analysing sub-systems, highlighting relevant systems in the environment, and for examining interactions between the various elements that have been identified.

3.6 Systems concepts: dynamic behaviour: control

A system does not usually behave in a random manner – its actions are governed in some way. This can be achieved by using the control models, either singly or in combination, shown in [Figure 32\(a\) and \(b\)](#). The feedback (or closed loop) control model in [Figure 32\(a\)](#) works as follows:

- a feature of the output from the transformation process is monitored;
- information regarding this is 'fed back' to a comparator process that compares the value of the feature with a predetermined 'objective' value;
- if the value falls within the predetermined limits, no further action is taken;
- if the value is outside the predetermined limits, a signal is sent to alter an input to the process in order to correct its operation.

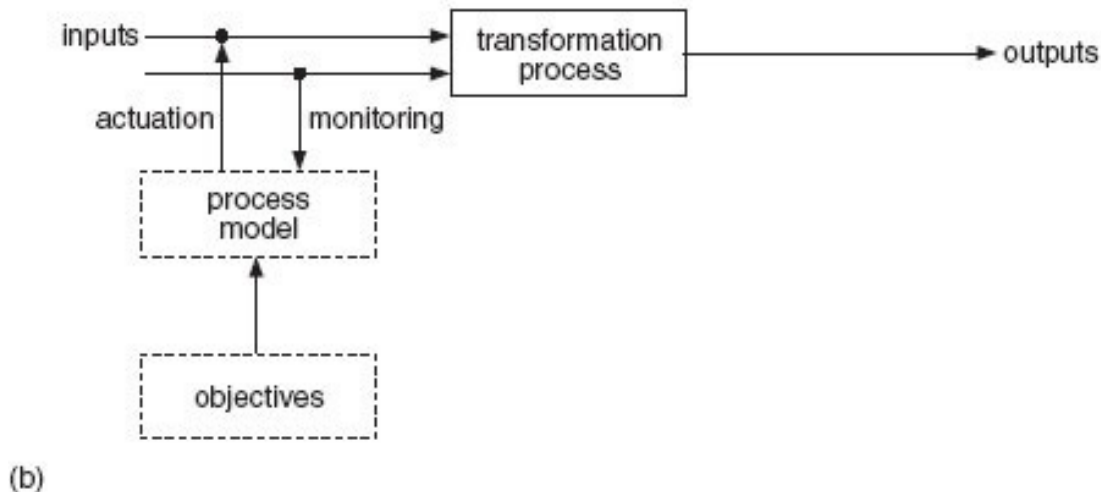
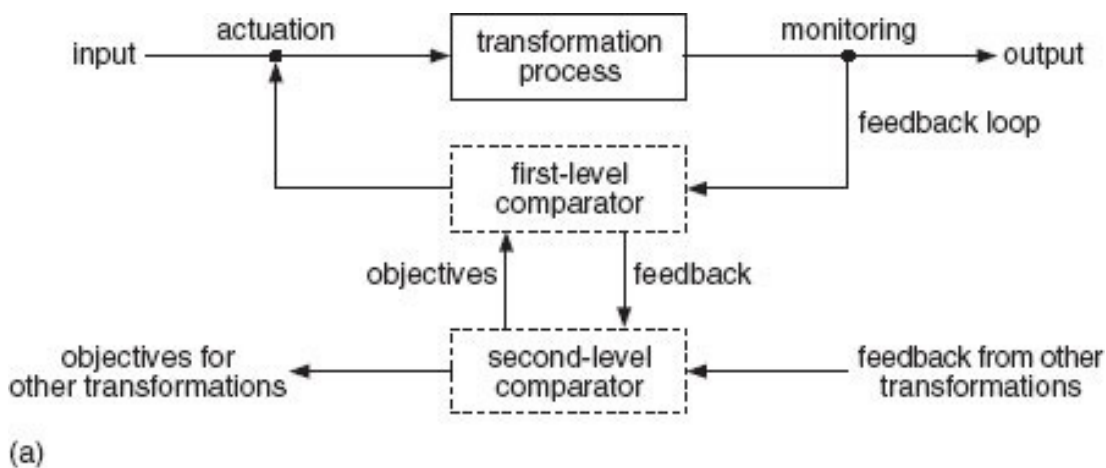


Figure 32 (a) Feedback and (b) feedforward control

For example, suppose that the transformation process is drying in an oven. The objective is to maintain the temperature of the oven to within $\pm 8^\circ\text{C}$ of a particular value. A sensor monitors the temperature of the oven. This is then compared with the preset limits, and corrective action is taken through an actuator if necessary. The important features of the model are as follows:

- It repeats the basic structural pattern of input-process-output, with the addition of a feedback loop.
- It shows two different levels of control: the first deals with feedback directly from the transformation process and the second with feedback from the lower level. This pattern of hierarchy of control is a feature of nearly all operations systems. The function of the higher level system is to provide standards for those at a lower level and to take corrective actions that are not within the competence of the lower level. The hierarchy is not, of course, restricted to two levels.
- All or part of the output of the transformation process is monitored, either continuously or at intervals.
- Information from the monitoring sensor is fed back to a comparator, which, as its name suggests, compares the value with a predetermined standard. If the value falls within the permitted range, no further action is taken. If the value is outside the limits, two courses of action are open. If the deviation can be corrected at that level of the system, an actuating signal is sent to alter one of the inputs to the process. If the deviation is outside the competence of this control level, the occurrence is reported to the higher level of control. The comparator may be a physical device (a microcomputer, for example), a person or a committee – though that would be rare at this level.
- Actuation involves altering one of the inputs to the system to create the desired change in output.

The feedforward (or open loop) control model in [Figure 32\(b\)](#) is a control strategy used to correct or compensate for the variance of an input, to achieve a desired output variable. Feedforward can be used when the relationships between input, process and output are so well understood that it is possible to predict the effect of a particular change to an input. Feedforward control involves monitoring the condition of a critical input variable and predicting, using a model of the process, the effect that a variation will have on output. This approach means that changes to the measured input or some other input variable can be made 'before the event'. For example, suppose a strip is to be rolled to a thickness of 2 millimetres. The thickness of the incoming material could be monitored and roller pressure increased or decreased as necessary.

Particular attention has to be paid to the nature of the control parameters that are used as objectives in a control loop. For example, measuring machine hours and utilisation may, in the first instance, seem like a good idea. But if this results in the overproduction of components for stock it may be detrimental to the effectiveness of the system as a whole. This is another example of the importance of taking a holistic view.

The final concept associated with control is lag. This can be defined as the length of time that the control loop takes to correct an out-of-tolerance condition and can be expressed as:

$$\text{Mean control lag} = \text{Mean monitoring interval} + \text{Monitoring time} + \text{Feedback (2) + Comparison time} + \text{Feedback (1) + Actuation time}$$

The length of the lag in the control loop can have a significant influence on system performance. In a fully automated control system, total reaction time can be very small, but where clerical procedures are involved it may be lengthier.

[Figure 33](#) shows an operations system consisting of five linked processes involved in the manufacture of printed circuit boards (PCBs) and the time (in seconds) that each operation takes.

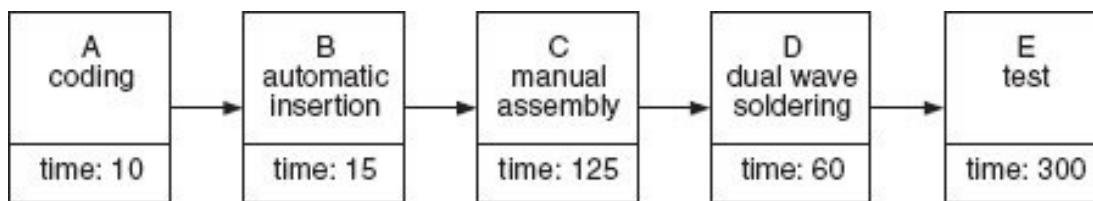


Figure 33 PCB operations system

Each operation in [Figure 33](#) involves:

A: Coding. The empty board is coded with part number and serial number.

B: Automatic insertion or mounting. Components are manually loaded and unloaded into an automatic mounting machine.

C: Manual assembly. Not all components can be mounted automatically. For example, block capacitors and switches have to be inserted into the PCB. The operator takes two boards and loads them into a carrier. The components are mounted. The fully populated board is mounted onto an in-feed conveyor that takes it through the soldering machine.

D: Soldering.

E: Testing.

The policy is to test one board fully every 10 minutes. In practice, depending on how busy the test technicians are with other work, the interval varies between 8 and 14 minutes, with the average at about 10 minutes. If a fault is detected, further tests are carried out to discover whether the problem is isolated or persistent. The corrective action taken depends on the type of fault and where it occurred. When problems occur with the settings on the automatic mounting machine they take about 10 minutes to correct from first detection during testing.

SAQ 12

If a fault occurs due to the settings on the automatic mounting machine, how long is the total mean control lag?

Answer

The mean control lag in seconds is calculated as follows:

Automatic insertion	15	
Manual assembly	125	
Dual wave soldering	60	
Average monitoring interval	600	
Testing	300	
Time taken to correct settings	<u>600</u>	
Mean control lag	1700	≈28 minutes

At a rate of four boards a minute, this delay could represent 112 faulty PCBs.

3.7 Systems methodologies for managing change

The use of systems concepts and models forms part of a process of investigation that is often described in the literature of systems, design and decision-making as a 'methodology', where a methodology is a process of enquiry, not a method to produce a predetermined result.

A systems methodology has the following characteristics.

- It is, or it provides, the means for the investigator to draw up a plan for studying a situation. This encourages important avenues to be explored and considered.
- It is not a simple checklist of actions that will lead to the 'right' answer; instead, it poses open-ended questions that can be answered in a variety of ways.
- It is generally iterative; stages may well be passed through several times before the investigation is complete.
- It enables a statement to be made of the objectives of the study and allows these objectives to be reviewed and modified.
- It provides guidelines for tackling problems, based on proven experience with similar types of problem.
- It provides team members and others with a common language for discussing the project.
- It is a framework for setting objectives, timescales and cost targets for the project.
- It makes it easier to manage project progress, identify problem areas and put things right.

Systems methodologies developed from operational research work during the Second World War and from subsequent work on technical and social problems by the RAND Corporation and others in the United States. Since then, systems engineering methodology has developed. Today there are two main generic variants:

- the hard systems approach – used when problems and opportunities can be clearly defined
- the soft systems approach – used when there is little or no agreement about the problem.

The development of systems engineering methodologies will be discussed in detail in the next section but I will outline briefly the two major generic variants here.

3.8 Systems methodologies for managing change: hard systems approach

The stages of the hard systems approach are illustrated in [Figure 34](#) and simplified in [Figure 35](#). The model shown in these figures was developed by the Open Systems Group from earlier work by de Neufville and Stafford (1974). The stages 'problem/opportunity' and 'implementation' are shown in solid boxes because they occur in the real world. The other stages are in dashed boxes to indicate that they are thought processes.

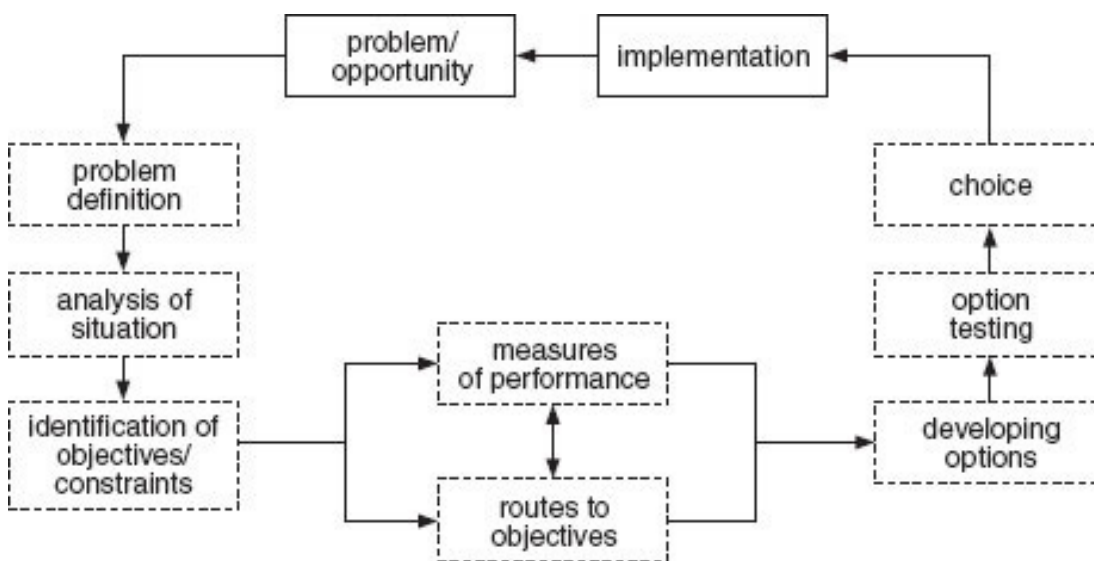


Figure 34 The stages of the hard systems approach

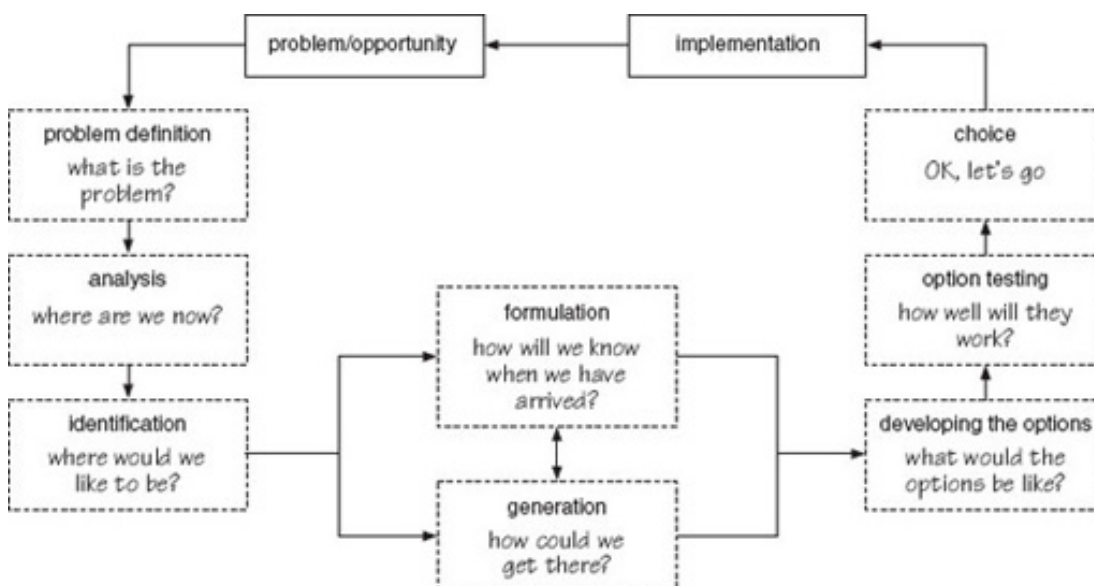


Figure 35 The hard systems approach simplified

For the sake of diagrammatic simplicity, two features of the hard systems approach are missing from [Figures 32](#) and [33](#). First, *iteration* between stages is not shown. The stages of the approach are shown occurring in a logical sequence – this orderliness is the essence of the methodology. Jumping from one stage to another in a haphazard sequence is to be avoided. But there is always the possibility, and in some cases the necessity, of referring back to an earlier stage and then repeating subsequent steps. This iteration can occur for two reasons:

- subsequent work has uncovered a mistake in earlier reasoning or an area of uncertainty
- something happens in the environment of the investigation that makes previous assumptions invalid.

The second feature missing from the diagram is *agreement*. After each stage, it is essential that all stakeholders:

- agree that the work has been carried out correctly
- agree with the aims, content, timescale and people involved in the next stage.

Each stage is described briefly below.

Stage 1: Problem definition (what is the problem?)

The aim of the first stage is to identify and describe the problem or opportunity. While each stage depends on the success of the previous stage, it is the initial stages of a project that set the direction for the work as a whole. For this reason a clear definition and firm agreement on the problem or opportunity are essential.

Problems and opportunities are like two sides of a coin: one of them can always be formulated in terms of the other. The best way to distinguish between them is to think of:

- a problem as an unwelcome deviation from a standard or expected state of affairs – its solution involves the restoration of the existing, satisfactory state of affairs
- an opportunity as a chance to improve on the existing state of affairs.

Stage 2: Analysis of the existing situation (where are we now?)

Having defined and agreed on the problem, it is necessary to decide on the system in which you consider it plays a part. In practice the two stages are closely linked and the analysis of the existing system nearly always means a redefinition or refinement of the problem or opportunity. Identifying and defining the problem and the system or systems that relate to it are critical for the success of subsequent analysis.

Stage 3: Identification of objectives and constraints (where would we like to be?)

This stage forces the project team to make explicit the objectives and constraints associated with the problem or opportunity. This is valuable for several reasons.

- It forces everyone concerned to clarify what they hope to achieve.
- The need to agree objectives and constraints can bring into the open disagreements that otherwise might emerge only at a later stage of the approach.
- The process of defining, elaborating and agreeing objectives and constraints helps to maintain commitment to the project.
- It lays the foundation for subsequent activities, since once the objectives and constraints have been defined it is possible to decide what is needed to achieve or avoid them.

Objectives and constraints can be quantitative or qualitative.

Stage 4: Generation of routes to objectives (how could we get there?)

This stage explores the different ways of achieving the defined objectives. It is the most imaginative and free-thinking stage of the approach. The idea is initially to generate as many ideas as possible, then to whittle the list down to two or three 'definite possibilities' that can be carried further in the development stage.

Stage 5: Formulating measures of performance (how will we know when we have arrived?)

The hard systems approach emphasises the need to have measurable means of assessing the efficacy of any potential solution or design, but recognizes that this may not always be possible.

Stage 6: Developing the options (what would the options be like?)

The objective here is to develop the routes to objectives generated in Stage 4 to the position where they could be implemented if the decision to go ahead were given. This involves doing sufficient work on each option for technical and other details to be defined, and for costs and benefits to be assessed, and for a sound decision to be taken, while at the same time minimising the time and resources devoted to the task.

Stage 7: Option testing (how well will each work?)

While the identified objectives and constraints have been referred to constantly during the development stage, the testing stage of the approach is a more formal analysis of each option. Its objective is to determine whether:

- the option will meet the operational objectives
- it is technically feasible
- it is organisationally feasible
- it will meet the financial objectives.

Stage 8: Choice (OK, let's go)

You might imagine that after all that has gone before, the decision about whether to go ahead or not would be automatic, but this is rarely the case. There will still be much discussion and 'fine tuning' necessary to ensure that the proposal is acceptable. It is at this stage that any qualitative measures of performance are brought into play.

Implementation

Implementation involves all the detailed design, development and installation tasks required to get the agreed proposal operating.

[Figure 34](#) shows an arrow leading from 'implementation' to 'problem/opportunity'; this recognises that implementation is never the end of the story. The successful completion of a project will give rise either to other opportunities or to a further set of problems that need to be tackled.

3.9 Systems methodologies for managing change: soft systems approach

Various 'softer' approaches to problem solving have been proposed. The one that I shall describe is based on (although not exactly the same as) the methodology developed by Peter Checkland and his collaborators at the University of Lancaster. This has been applied to systems problems in a number of projects.

The soft systems approach is based on a number of key principles.

- Problems do not have an existence that is independent of the people who perceive them.
- Solutions are what people perceive to be solutions.
- People perceive problems or situations differently because they have different beliefs about what the situation is and what it should be.
- Problems are often linked into what are called 'systems of problems' or 'messes'.
- The analyst, researcher or manager trying to solve the problem is an integral part of it!

The stages of the soft systems approach are illustrated in [Figure 36](#).

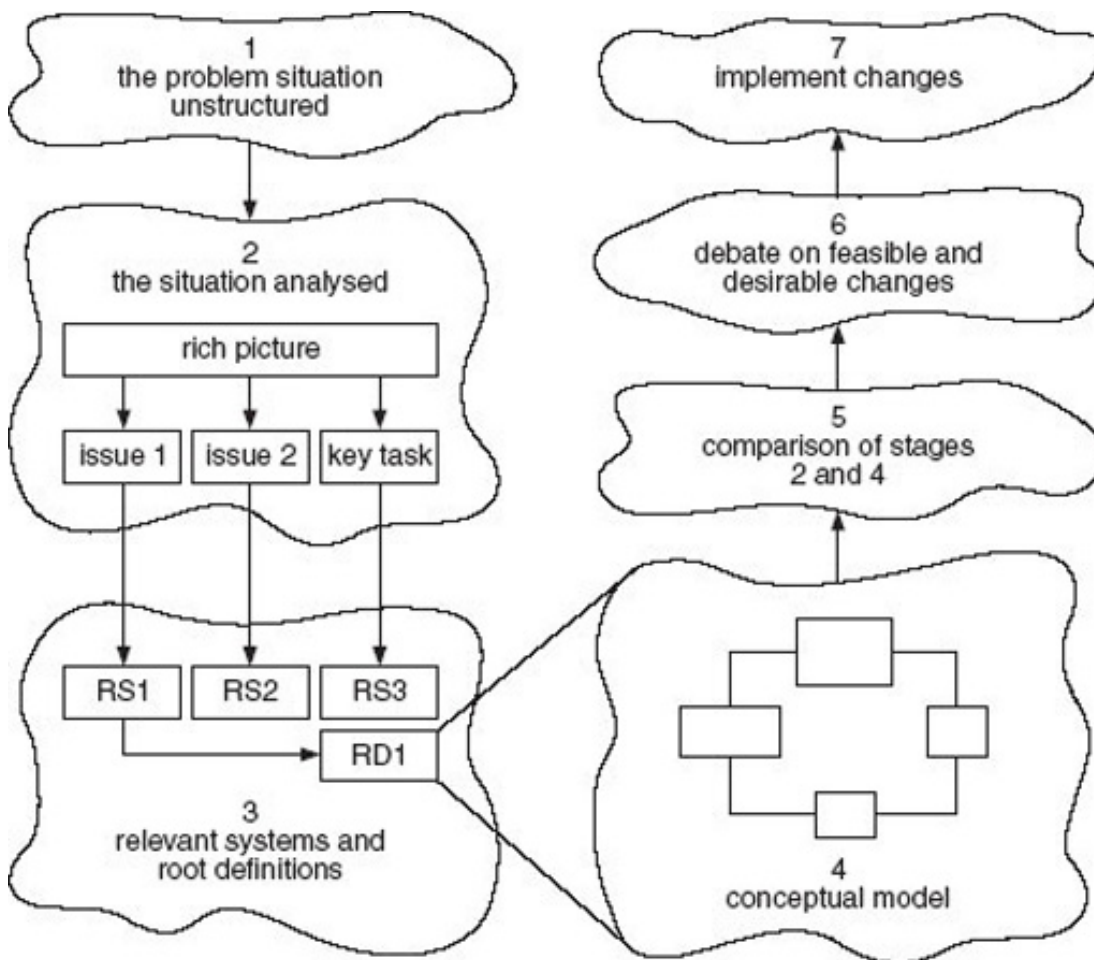


Figure 36 The soft systems approach

Stage 1: The problem situation unstructured

The approach begins with a situation in which one or more people perceive that there is a problem. It will not be possible to define the problem or its setting with any precision and, in any event, the different people involved will have different ideas.

Stage 2: The situation analysed

The first step is to develop a picture (called in soft systems terminology a *rich picture*) that encapsulates all the elements that people think are involved in the problem. Once the rich picture has been drawn, the analyst will attempt to extract 'issues' and key tasks.

Issues are areas of contention within the problem situation. Key tasks are the essential jobs that must be undertaken within the problem situation.

SAQ 13

The essence of rich pictures is that they represent all the elements in a situation in an unstructured way. By all the elements I mean both 'hard facts' (if there are any) and the softer, more subjective aspects of the situation being considered. Look at the rich picture of a new product introduction process shown in [Figure 37](#), then answer the questions.

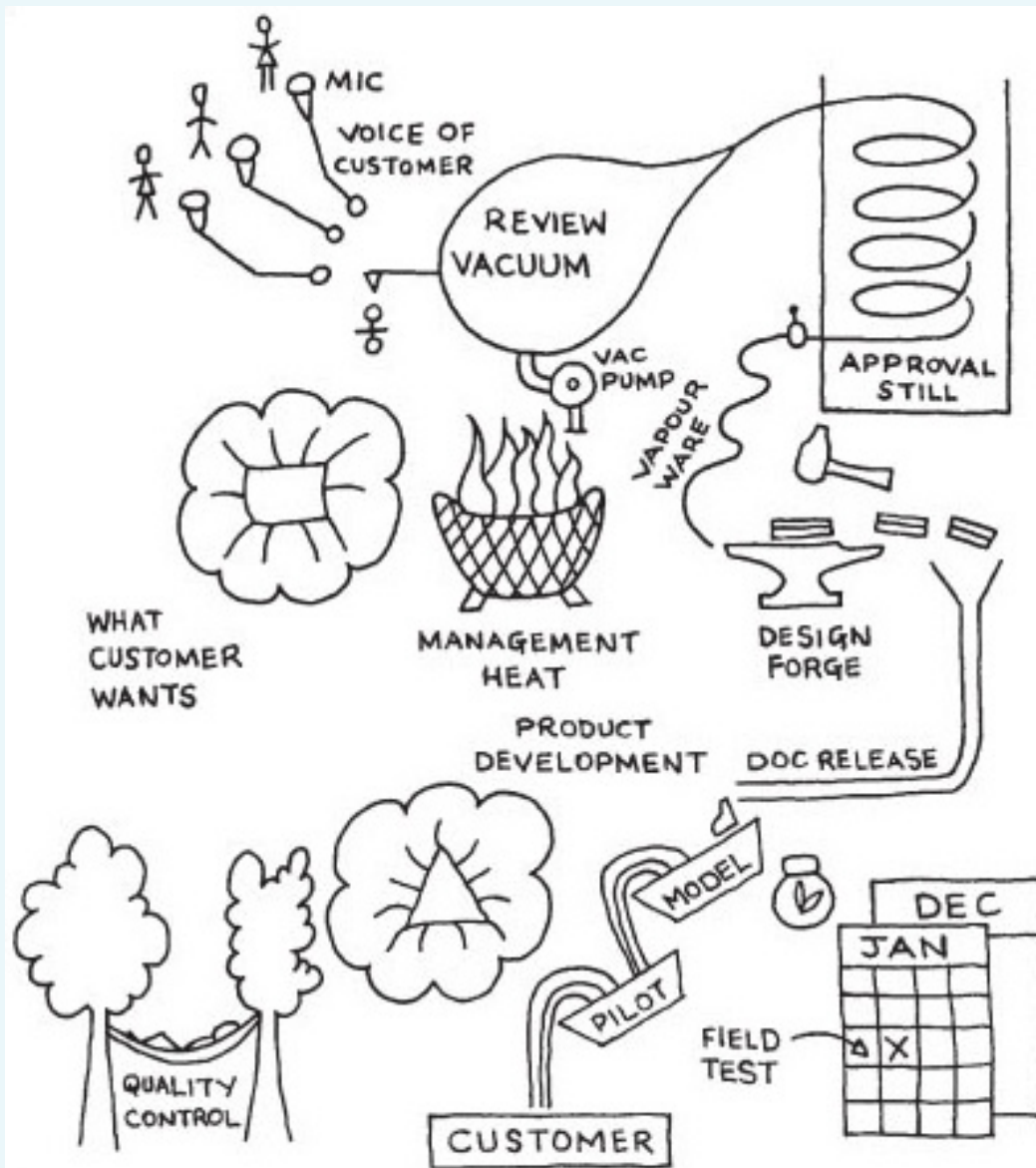


Figure 37 The new product introduction process

- (a) What situation do you think may have led to the drawing of the rich picture?
 - (b) What are the hard elements in the situation?
 - (c) What are the soft elements in the situation?
 - (d) Briefly write the 'story' that the rich picture tells.
-

Answer

- (a) The situation that led to the rich picture being drawn might have been a dissatisfaction with the new product introduction process in terms of its ability to deliver what the customer wants or the time that it takes.
- (b) The hard elements in the situation could be:
 - customer wants
 - product as delivered
 - dates of teststhough all these have 'softer' elements.
- (c) The soft elements are:
 - management heat under the review vacuum
 - quality control sleeping peacefully
 - the gap between the voice of the customer and customer wants
 - the convoluted 'pipe' of management approval.
- (d) The story that the picture tells is one in which the voice of the customer is elicited and defined, albeit with imperfect transmission of requirements. The requirements are then reviewed with management urging action leading, perhaps, to premature decisions about requirements. Approval takes a long time and results in 'vapourware' that influences the design process. Documents are released from design into product development, which produces models and undertakes field tests and pilots. The product is then delivered to the customer but is not the same as what was wanted. And all the while, quality control slumbers on.

Stage 3: Relevant systems and root definitions

The issues and key tasks extracted from the rich picture become the basis for defining what are called the 'relevant systems'. For example, suppose the problem situation is a deteriorating performance in a call centre. One of the issues might be the (high) turnover of call centre operators. This might lead (depending on the point of view taken) to an idea of the call centre as an 'employment-providing system' or an 'entertainment system'. There is no reason to restrict relevant systems to one issue or key task. Further analysis of more than one relevant system might provide valuable insights into different perspectives on the situation.

The initial idea of the relevant system is then expanded into a root definition. The root definition of the 'entertainment system' might be, 'A system to ensure that call centre operators find their jobs as interesting as possible and that the social and physical office environments are stimulating.'

Stage 4: Conceptual model

The conceptual (or activity) model contains all the activities that the relevant system would have to perform. The model is usually drawn as a block diagram.

Stage 5: Comparison of Stages 2 and 4

The objective of the comparison stage is to relate the conceptual model to the problem situation as depicted in the rich picture. The idea is to highlight differences between the two so that potential improvements to the problem situation can be identified.

Stage 6: Debate on feasible and desirable changes

The comparison undertaken in the previous stage can have two results.

- It can cause opinions to change on the problem situation and the issues arising from it.
- It can provide an agenda for change.

In either case (though both may result), the objective of this stage is to debate, with all concerned, the changes proposed to ensure that they are both desirable and feasible. The aim is to arrive at consensus about the proposed changes. However, as the soft systems approach specifically attempts to explore and reconcile different views and perceptions of the problem situation, this may not be totally achievable. Thus, soft systems seek accommodation between these different views to enable action to be agreed.

Stage 7: Implement changes

Finally, the agreed changes are implemented.

Like the hard systems approach, soft systems methodology is not seen as a 'one pass' procedure, but as a learning process. Iteration is a feature of the methodology's application. Learning is achieved in both approaches by the use of models, although soft systems has subsequently been enhanced to include a specific analysis of the culture and politics of the problem situation, as shown in [Figure 38](#). This is important if changes are to be culturally feasible and politically acceptable. Combining these two approaches in soft systems leaves us with a picture of it as a learning system, as shown in [Figure 39](#).

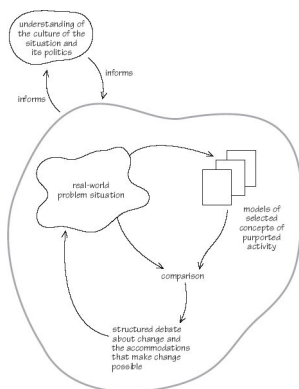


Figure 38 Cultural aspects within the soft systems approach

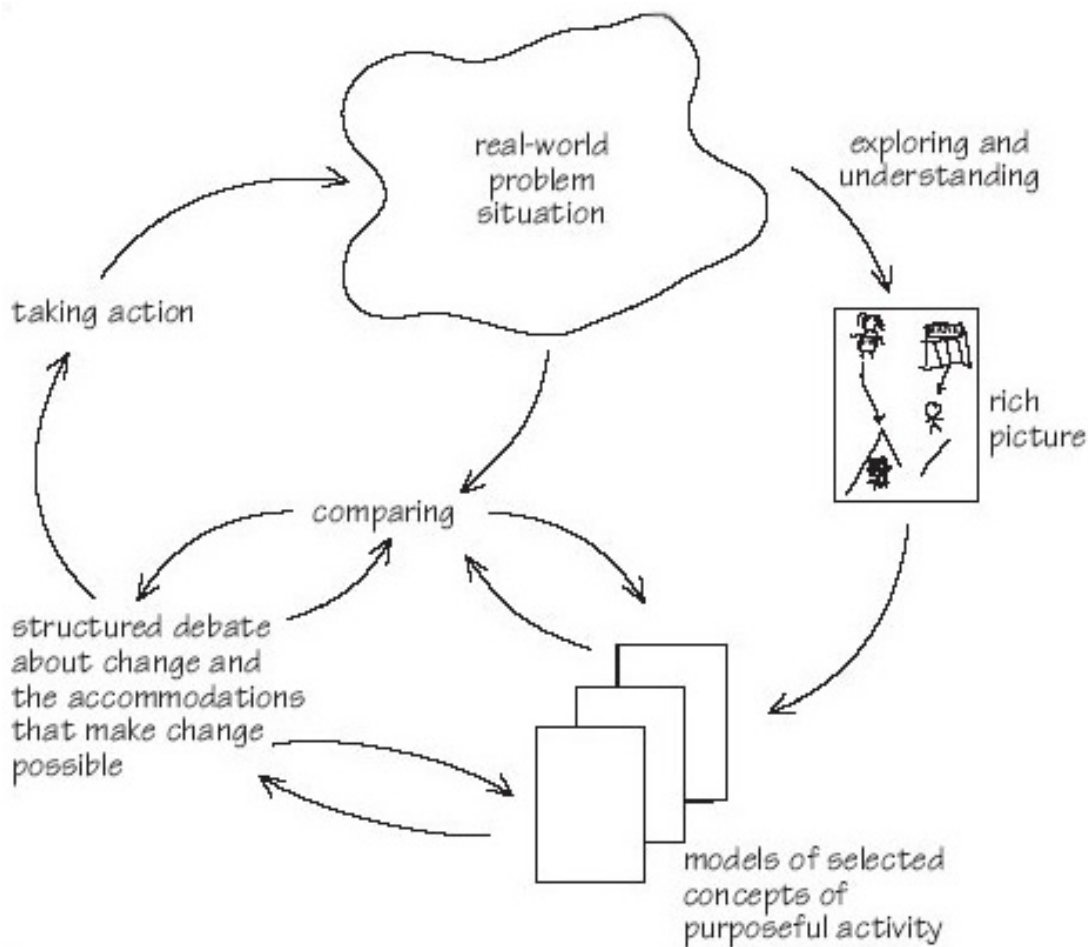


Figure 39 Soft systems methodology as a learning process

3.10 Systems techniques

The two systems methodologies provide a framework for the application of problem solving, analysis and design techniques. These fall into three groups.

- **Diagramming:** ranging from single systems maps to complex flow charts. Diagrams of one sort or another provide a method of analysis, design and communication.
- **Modelling:** simulation is used extensively to analyse the dynamics of an existing system and to predict the behaviour of a proposed one. Financial models are created to justify expenditure.
- **Creativity:** solving problems, large or small, requires creative thought. Various creativity techniques are commonly used in systems work.

3.11 Summary

This topic has introduced the systems approach, which is the foundation of systems engineering. The systems approach consists of three elements.

- A set of concepts that can be used to understand the structural and dynamic features of operations systems.

- Methodologies for managing change. Two current methodologies have been presented: the hard systems approach can be applied in situations where there is a measure of agreement about the problem to be addressed and the soft systems approach is used where there is no agreement about the nature of the problem.
- A set of techniques that are applied within the context of a methodology.

The systems approach in various forms will be used, either explicitly or implicitly, as the basis for all of the other material in this course.

4 What is systems engineering? The career of a concept

4.1 Beginnings

Systems engineering has its roots in three linked strands of thinking: the concepts of systems science, engineering and public policy problem resolution. The first of these can be traced back to the work of von Bertalanffy (1968, pp. 8–15, 96–98) and others during the 1920s and 1930s but received a significant impetus when, in 1954, the Society for General Systems Theory was established at the annual meeting of the American Association for the Advancement of Science. The society later changed its name to the Society for General Systems Research and launched an ambitious programme:

The Society for General Systems Research was organized in 1954 to further the development of theoretical systems which are applicable to more than one of the traditional departments of knowledge. Major functions are to: (1) investigate the isomorphy of concepts, laws and models in various fields, and to help in useful transfers from one field to another; (2) encourage the development of adequate theoretical models in the fields which lack them; (3) minimize the duplication of theoretical effort in different fields; (4) promote the unity of science through improving communication among specialists.

(von Bertalanffy 1968, p. 13)

As this quotation suggests, the perspective of the Society for General Systems Research was *scientific*, but in parallel the use of systems as a *technology* for problem solving was developing rapidly.

In the 1930s the American pragmatic philosopher John Dewey proposed that there should be five phases or aspects of what he termed 'reflective thought':

- suggestions – in which the mind leaps forward to possible solutions
- an intellectualisation of the difficulty or perplexity that has been felt, into a problem to be solved
- an evaluation of the suggestions in turn as a leading idea or hypothesis, to guide observation and the collection of factual material
- elaborating the ideas through the application of reasoning
- testing the hypothesis by overt or imaginative action.

The five phases of Dewey's reflective thought (Dewey, 1933) exhibit the characteristics of systems methodologies. The focus is on a problem that needs to be addressed. The approach involves investigation and data gathering to test the efficacy of different possible solutions. Dewey's approach contains the two key elements of systems engineering: the advocacy of a method of enquiry and action, and the application of the method to the solution of real-world problems.

4.2 The use of systems analysis in public policy

The application of mathematical techniques to military operations was pioneered in Britain during the Second World War (see Box 7) and became known by a variety of names (Hoos, 1972, p. 42). At the end of the war, the United States Air Force sponsored the application of those techniques and methods to problems of US national security. Funds

to investigate the effects of new weapons systems and for the exploration of defence policy issues were allocated to defence contractors. From one of these, the Lockheed Aircraft Corporation, the RAND (**R**esearch **a**nd **D**evelopment) Corporation was spun off as a non-profit-making advisory consultancy that initially concentrated on issues of public policy. As RAND's internet site puts it:

From its inception in the days following World War II, RAND has focused on the nation's most pressing policy problems. High-quality objective research on national security became the institution's first hallmark. In the 1960s, and in the same spirit, RAND began addressing major problems of domestic policy as well.

Today, RAND researchers operate on a uniquely broad front, assisting public policy makers at all levels, private sector leaders in many industries, and the public at large in efforts to strengthen the nation's economy, maintain its security, and improve its quality of life. They do so by analyzing choices and developments in many areas, including national defense, education and training, health care, criminal and civil justice, labor and population, science and technology, community development, international relations, and regional studies.

(RAND, 2000)

This quotation suggests three important features of the RAND approach. It had its origins in military systems and policy issues. It spread from this origin to be applied in a wide range of contexts. At the core of the RAND approach is the rational analysis of the choices that are available to decision-makers.

Gradually, the accepted name for this type of policy analysis became 'systems analysis', which an early practitioner denned as:

... an inquiry to aid a decision-maker choose a course of action by systematically investigating his proper objectives, comparing quantitatively where possible the costs, effectiveness, and risks associated with the alternative policies or strategies for achieving them, and formulating additional alternatives if those examined are found wanting.

(Quade, 1967)

Box 7 The origins of systems analysis

The systems approach is a lineal descendent of, and shares a common heritage with, operations research. As it relates to the man-machine relationship, the technique dates back to the Industrial Revolution, if not earlier. For military planning specifically, it emerged in its present form during World War II, when the British High Command sought the help of teams of physicists, biologists, mathematicians, and other specialists to devise strategy for incorporating advanced and *unconventional* equipment into the air defense system. The new weapons and weapons systems, of which radar was an early example, were so radically different in concept from anything previously known that traditional military experience had no relevance. The new methods of operations analysis which were developed formed the core of the technique called 'operations analysis' at the time. Subsequently, as it was refined and expanded, it came to be known as, or was almost indistinguishable from, operations research, systems engineering, management science, cost-effective analysis, and systems analysis.

Source: Hoos (1972, p. 42)

Systems analysis and its associated techniques were widely and enthusiastically adopted by public bodies and commercial companies alike. Central to the approach was the idea of rationality based on financial or operational modelling. And it was this characteristic that caused systems analysis to come under critical attack from those who objected to what they regarded as the dilution of what should be a political process (Hoos, 1972; Lilienfeld, 1978) or who felt that rational analysis was not an adequate basis for decision-making.

Now read the following case studies:

- Fire Department Deployment Analysis – Modelling in Public Policy, which provides an example of an early RAND analysis carried out in an attempt to improve the deployment of resources.
- The Roskill Commission – Using Cost-Benefit Analysis for Public Policy, which discusses the problem associated with the application of rational methods to complex planning and decision-making processes through the example of the lengthy inquiry held during the 1970s and 1980s.

Please click on the 'View document' link below to read case study 1 - Fire Department Deployment Analysis – Modelling in Public Policy.

[View document](#)

Please click on the 'View document' link below to read case study 2 - The Roskill Commission – Using Cost-Benefit Analysis for Public Policy.

[View document](#)

As the example given in the London's Third Airport paper indicates, there were perceived problems associated with the use of systems analysis in public decision-making. These were as much to do with unrealistic expectations and the technological optimism of the time, as they were with faults in the approach. The aim of the Roskill Commission had been to avoid 'arbitrary and subjective judgements' and it had attempted to include the 'indirect impacts' of the new airport. However, far from improving the acceptability of cost-benefit analysis, attempts to broaden its scope merely attracted the criticisms that the analysis had omitted important factors, that the broadening of scope had not gone far enough and that the valuations that had been incorporated into the process were wrong. Attempts to answer these criticisms only made matters worse. In accepting that 'the cost-benefit analysis could never include all the factors relevant to the decision' the Commission was being realistic, but at the same time weakened the methodological basis of its decision. It provided the pretext for one of its number, Colin Buchanan, to dissent from its decision and, in doing so, opened the door to the political opposition that followed. An important statement in the example in London's Third Airport is, 'The forces of economic reason may have declared for Cublington, but the forces of environmental emotion were in favour only of Foulness, and they proved far stronger both in number and intensity.' The mistake was in expecting, initially, that cost-benefit analysis would provide 'the answer'. Then that it could be used as the basis for rational judgement, instead of regarding it as merely one of a number of inputs to a wider, essentially political, decision-making process.

The airport was never built at Maplin Sands or at any of the other sites considered by the Roskill Commission, but the controversy that focused on cost-benefit analysis called into question the use of systems analysis in matters of public policy. It was never to recover fully and, although the work of the RAND Corporation continues to this day, the glory days of systems analysis, in this sense of the term, were over.

4.3 The use of systems engineering in organisations

The development of systems engineering was contemporaneous with that of systems analysis in public policy. Though its origins can be traced back to the 1930s and 1940s (Hall, 1962, p. 7), its more widespread application can be dated from the early 1950s. The earliest formal teaching of systems engineering was a course presented in 1950 at the Massachusetts Institute of Technology by G.W. Gilman, who was then Director of Systems Engineering at Bell Laboratories. Gilman was a strong promoter of the use of systems engineering at Bell Labs and was instrumental in developing training material that was used in company courses (Hall, 1962, p. vii).

One of the Bell Labs executives, Arthur Hall, was closely associated with the introduction of systems engineering into the organisation, and considered that it was necessary because:

- products were increasing in complexity
- the needs of consumers were expanding
- the business environment was expanding rapidly in terms of markets, technology and concerns
- there was an acute shortage of technically and scientifically trained people.

It is interesting to note that, half a century later, the argument has not changed a great deal. Stevens et al. (1998, p. 2) state that:

Systems engineering is the 'key technology' to manage complexity created by:

- increased complexity of products
- globalisation of the market-place
- the erosion of trade barriers
- reduction of product development cycles
- software as the dominant force for change in almost all new products
- worldwide deployment of new technology in ever shorter timescales
- systems constructed from bought-in technology and components
- reuse of components, information and knowledge across projects
- partnerships for product development leading to worldwide teams
- the transition from paper-based control to control through electronically managed information
- an understanding that intellectual capital often is the major part of the assets of a modern organisation.

Question 4

What are the main differences between Hall's justification of systems engineering and that provided by Stevens et al.?

.....

Answer

Perhaps the main difference is the emergence, over the last half-century, of the importance of computing, which features in one way or another in three of the items on the list of Stevens et al. A second difference is that Stevens et al. regard 'time', or the lack of it, as a significant factor leading to increased complexity. Hall cites

shortages of technically and scientifically trained staff. Essentially, however, both justifications are that 'things' are getting more complex.

The five-stage systems engineering methodology adopted in Bell Labs consisted of:

1. **Systems studies: programme planning**

During this phase a comprehensive range of environmental factors is investigated with the aim of laying out a possible broad development programme for the organisation. Senior managers can then make an informed choice of which projects to accept and the amount and type of resources to devote to them. A secondary aim is to increase the knowledge in the organisation of trends and developments in its business environment.

2. **Exploratory planning: project planning 1**

This phase of the methodology is focused on a single project. This may be one of the possibilities laid out in the programme defined in the previous stage, or the systems engineering methodology may start at this stage if the need for the project has been separately defined.

Exploratory planning starts with analytical work to define the problem, or need, that the project is addressing. The second step is then to select the objectives of the system in relation to the problem or need that has been defined. The objectives will guide the subsequent selection of alternative solutions and the associated analyses.

The next step, termed systems synthesis, is concerned with compiling or devising alternatives that might satisfy the objectives that have previously been defined. Each alternative is then tested during the systems analysis step. The results of the testing are used during the selection stage to choose the best solution. Finally, the results of the work during this stage are communicated to senior management and others having an interest in the project, with a recommendation to go ahead, do no further work, or that more investigation is required.

3. **Development planning: project planning 2**

If the decision is to proceed with the project, the systems engineering team would then produce a detailed plan which would be used to guide subsequent work.

4. **Studies during development: action phase 1**

The project is then handed over to development engineers responsible for undertaking the detailed design. The role of systems engineering during this stage is to assist engineers with any special studies that might be required during development and implementation.

5. **Current engineering: action phase 2**

This stage covers any follow-up work undertaken when the system is in use.

4.4 The use of systems engineering in organisations: different organisational arrangements

Hall identified three different organisational arrangements that might provide a framework within which systems engineering work could take place within the organisation. The first of these, which he termed the departmental form and regarded as the lowest level of arrangement, was essentially a temporary team of specialists brought together, under the management of a team leader, to undertake a specific project. The team consisted of

members of each of the specialist development departments and was what would now be called 'multifunctional'. It does not seem that the team was co-located since 'Problems uncovered by the team would [then] be brought back to the regular department for development' (Hall, 1962, p. 14). Hall seems to have been proposing a form of weak matrix organisation, with team members responsible to their departmental head and to the team leader. The team was dissolved on project completion.

Hall's second structural arrangement was the task force. In this form the organisation was divided into separate projects with service departments that the projects could call on. This was probably an early form of project organisation. Hall's third form was a mixture of the two. In Bell Labs, systems engineering was established as a separate department, headed by a vice-president, and had equal status to the research department and the development departments.

As part of AT&T, Bell Labs was a noted pioneer in management methods, and it cannot be assumed that it was representative of the use of systems engineering at that time. Nevertheless, the approach gained acceptance rapidly, and a flavour of the type of problem to which it was applied can be gained from the case quoted in Box 8.

Box 8 The development of colour television

At the outset, the systems engineers concerned with this complex project were faced with a series of basic conditions and problems. They had to establish first that there was a need for colour television. Then it was necessary to determine that colour television was both technically possible and economically feasible.

Subsequently, they had to consider the environment in which colour television would start, grow, and continue. This was a most important consideration since, for practical reasons, black-and-white television came first, creating a large and growing public investment in black-and-white television receivers. In this environment, the systems engineers concluded that the new colour system must be compatible with the existing black-and-white service. In other words, the colour system must be so designed that colour broadcasts could be received in monochrome on existing black-and-white receivers, while the new colour receivers could receive black-and-white as well as colour broadcasts.

In retrospect, the choice of a compatible system seems utterly logical, but it was not so in the earlier development period. In fact, an incompatible system urged by some in the industry was approved as standard by the government regulatory body. This move later had to be undone before further progress could be made towards establishment of a practical colour television service.

Another broad problem for the systems engineering team was the definition of technical specifications. Involved in this question were such considerations as balancing the requirements of human vision for picture detail and colour characteristics against the potentials of apparatus performance and the availability of channel space for broadcasting stations.

In the latter case, early analysis indicated a need for basic advances in technology and invention to fit picture information into a narrower frequency band than could be achieved by earlier straightforward communication techniques. It was evident, too, that new apparatus would have to be invented or developed – particularly the picture tube – for reproducing colour images. There was a clear need as well for broad experience in propagating the radio signals and transmitting colour programs.

As the work moved beyond these initial determinations into the more practical stages, a host of more detailed problems had to be solved. These related to practical

apparatus design, practical operation under broadcast service conditions, industry participation in determining the signal specifications for transmission, and approval of these specifications as standards by the Federal Communications Commission.

Finally, there came the problems of establishing the colour television service, expanding studios, interconnecting networks, installing transmitters, and creating programme production groups. In the final stage, too, arose the matters of selling colour to television advertisers, marketing the new receivers, and measuring public reaction.

Thus, from concept to fruition, the colour television system presents a 'text-book' example of the systems concept in action. All of these matters required full consideration by the systems engineering team, working toward the single defined objective.

Source: Engstrom (1957)

There were other pioneering and influential organisations that embraced systems engineering philosophy and practice. Notable among these were the US Department of Defense and NASA.

At about the same time, in the UK, the application of systems ideas was taking a different form. A group of consultants based at the Tavistock Institute in London developed the idea of looking at an organisation as what they termed a socio-technical system. Two of the main proponents of this approach stated:

In our earliest study of production systems [in coal mining] it became apparent that 'so close is the relationship between the various aspects that the social and psychological can be understood only in terms of the detailed engineering facts and of the way the technological system as a whole behaves in the environment of the underground situation'.

An analysis of a technological system in these terms can produce a systematic picture of the tasks and task inter-relations required by the technological system.

(Emery and Trist, 1960)

The socio-technical systems approach can be summarised in three principles:

- '... the basic unit for the design of socio-technical systems must itself be a socio-technical unit and have the characteristics of an open system.'
- the second aspect of systems design was the recognition that the arrangement of parts, and the principles on which that arrangement were based, were important principles
- the third principle was that '... higher levels of organization can be achieved only by the fuller use of the inherent properties of parts as co-determinants of positional values. [...] what this principle means is, quite simply, that the best design for any productive system will be that which not only allows that the goals of any subsystem, any part, embody in some manner the overall system goals ... and allows that any such part is self-managing to the point that they [sic] will seek to cope with external variances by firstly rearranging their own resources ...'

(Emery, 1981, pp. 387–88)

The work of the researchers at the Tavistock Institute was immensely influential, finding application in a number of progressive European companies such as Philips and Volvo. Its focus was shop-floor production processes rather than products and, as the above list suggests, it was based on principles rather than a methodology.

A sustained attack on systems engineering came through the development of the 'soft systems methodology' introduced in Section 3. The Department of Systems Engineering at the University of Lancaster was founded in the mid-1960s with initial funding from Imperial Chemical Industries, then the UK's largest industrial company.

ICI was realising at that time that the operation of a petrochemical complex, in which the output from some plants is the input to others, entails engineering the whole as a single complex system. This was different from building and operating a plant to make a single product, and it seemed to some forward-looking managers in ICI's then Agricultural and Heavy Organic Chemicals Divisions that it would be useful if this kind of problem were being addressed by some university-based group: hence the gift to Lancaster.

(Checkland and Scholes, 1990, p. 16)

At the forefront of the development of systems ideas was Peter Checkland, who was recruited to investigate the application of systems engineering methodology, which he characterises as being concerned with developing a system to satisfy a defined need (Checkland and Scholes, 1990, p. 17) to less well-defined problems. This was found to be problematic, and in response Checkland developed a soft systems methodology. This was advocated as a 'broad approach to examining problem situations in a way that would lead to decisions on action at the level of both "what" and "how" ' (Checkland and Scholes, 1990, p. 18). Checkland argues that ill-defined problems are the most common situation faced by managers and that they need first to determine *what* needs to be done. 'This means that naming a system to meet a need and denning its objectives precisely – the starting point of systems engineering – is a special case.' (Checkland and Scholes, 1990, p. 17)

4.5 Methodologies associated with information technology

I recently undertook an exercise in a company that manufactures various types of agricultural machinery. I was happily using the term 'system' in its general sense when a senior manager stated 'In this organization we use the word "system" to mean a computer system, for other types of procedure or activity we use "process".' This is a commonly encountered distinction and the evolution of computer systems, and particularly software, has been an important contributor to systems engineering.

During the period 1965–80 the first wave of mainframe-based computing swept through organisations, and associated with this came a host of new jobs. One of these was 'systems analyst'. The work of systems analysts was to:

- investigate the financial, technical and organisational feasibility of a computer application
- design and agree the elements, structure and processes involved in the new system
- specify the programs needed
- write procedures and instructions
- implement the new system.

In the ensuing years the word 'system' replaced earlier ideas of 'procedure' or 'method' in the lexicon of business improvement.

IBM, with its System 360 series of computers and later with the System 370 series, quickly came to dominate the market for mainframe-based computing. Confusingly, IBM called the people that it employed to help its customers with their applications 'systems engineers'. Increasingly, the two terms 'systems analysis' and 'systems engineering' were used interchangeably in the development of information systems. Two related terms can now be distinguished: software system engineering and software engineering.

The conscious development of software engineering, as opposed to the practice of developing software as a craft, can be dated from a remark made by F.L. Bauer to a working party of the NATO Science Committee and subsequently to a conference held in Garmisch, Germany, in 1968 (Bauer, 1993). The reason for this development was statements made to the Scientific Committee by Bauer:

In a sudden mood of anger, I made the remark, 'The whole problem comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process', and when I found that this remark was shocking to some of my scientific colleagues, I elaborated the idea with the provocative saying, 'what we need is *software engineering*.'

(Bauer, 1993)

The *software life cycle* illustrated in [Figure 40](#) can be contrasted with the *information systems engineering*-related views of systems engineering shown in [Figures 41](#) and [42](#).

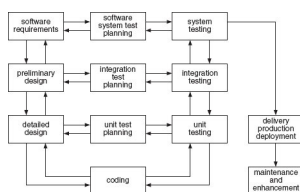


Figure 40 The software life cycle. Source: Faulk (1997)

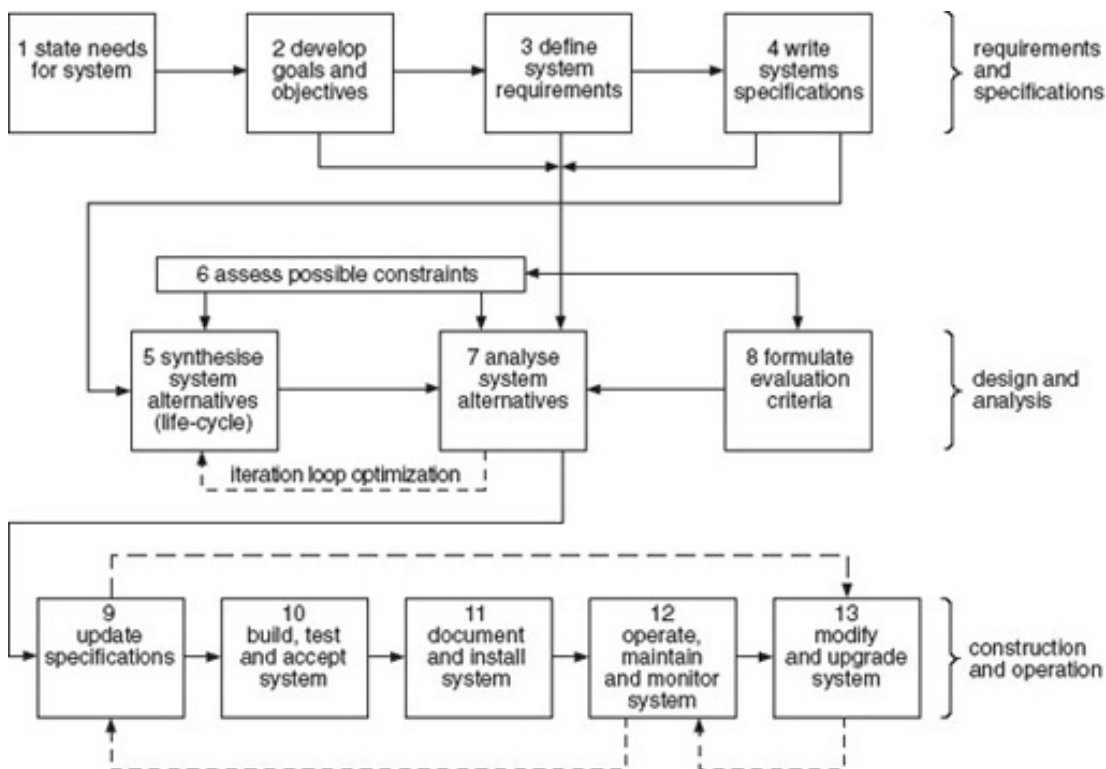


Figure 41 The generic systems engineering process. Source: Eisner (1988)

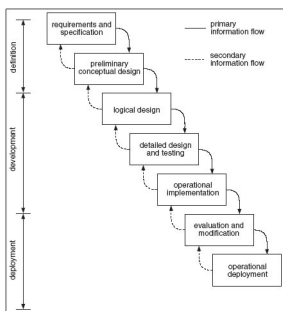


Figure 42 A waterfall model of systems engineering. Source: Sage and Rouse (1999)

In an attempt at unification, a fusion of the two types of model has been proposed (Sage and Palmer, 1989; Andriole and Freeman, 1993). The argument put forward is that although information systems engineering and software engineering have existed for some time as separate methodologies, several factors have combined to make a new, common approach appropriate:

- software-intensive systems are becoming more pervasive
- there has been a failure to learn from past experiences, usually of an unwelcome kind
- systems engineering is generic in character, being capable of application in diverse situations, but has the character of a 'meta-methodology' rather than being useful at a tactical, project level
- software engineering is, in contrast, focused completely on the software components of a system.

Andriole and Freeman (1993) propose a system life cycle that aims to integrate four major collections of system development activities: goal-setting, planning, design and construction. They argue that these activities take on a different character or specific meaning depending on the stage of development of the project. They represent the life cycle as the 'three-dimensional solid' shown in [Figure 43](#) and their explanation of the model is reprinted in Box 9.

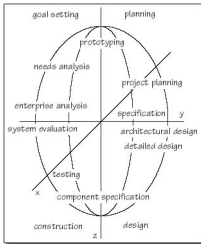


Figure 43 Andriole and Freeman's model of software systems engineering Source: Andriole and Freeman (1993)

Box 9 Andriole and Freeman's model

This life cycle bears a resemblance to Boehm's (1988) spiral model as a graphical device to represent the fact that the software systems engineer must make multiple passes through similar activities (concept development, prototyping, specification, design, implementation of the next level of technology). It also can trace some heritage back to Kerola and Freeman (1981) and others who first observed that system development involves a set of basic activities such as planning, design, implementation, and testing, all of which are in operation (to some extent) throughout the life cycle, being repeated with different emphases and contexts.

Looking down the time axis, the four quadrants represent four major collections of system development activities: goal setting, planning, design and construction. Each of these is an abstraction for a myriad of development activities, some of which span two quadrants. Depending on the position on the time axis, these activities take on different specific meanings. For example, needs analysis (labelled here as a 'goal setting' activity) at early stages of development is concerned with understanding the needs of the entire context in which a proposed system will be embedded (thus involving potential owners, clients, users and operators of the system). At later stages of development, however, the needs analysis activities might be concerned with determining the needs of a VDT operator (thus involving psychologists, terminal operators and medical experts).

The following definitions are intended to provide a non-exclusive outline of the activities that might be found in each quadrant.

1. Goal setting includes enterprise analysis, mission 'planning', needs analysis and most of the activities often associated with requirements engineering. Goal setting is front-end intensive and oriented to most aspects of the initial requirements analysis process.
2. Planning includes those activities that focus on or directly affect a large amount of development work, until the next planning phase appears in the life cycle. This includes prototyping, which provides fundamental information to help us decide how to proceed, project planning, which lays out detailed work

schedules, and specification activities, which provide the detailed 'work orders' for future work.

3. Design includes those activities that devise the structures (organizational, hardware of all sorts, software, procedural) which, when implemented, will provide the suitably constrained functionality needed to meet the goals. We differentiate between high-level (architectural) and detailed design. Crossing into the construction realm are those activities concerned with individual components. In the case of software, this is the detailed programming activity.
4. Construction activities include the various levels of testing which range from unit or component testing to acceptance or field testing. The various activities involved in measuring a system's performance and effects are included in this quadrant.

In the three-dimensional solid described by this life cycle, different fields of expertise and different people can be seen to touch different quadrants at different levels (times). The volumes thus described are, of course, highly irregular. For example, psychology (in a broad sense) might come into play in the early stages of goal setting by providing an insight into the workings of a large organization and the potential effects of a new software-intensive way of doing business; at a later stage by providing detailed parameters for VDT formats; and during operation of a system, by providing us with the methodology for evaluating the effect of a system on its clients.

Progress over time in creating, operating and modifying a software-intensive system is represented by a spiral path downward through the solid. Although such a path can represent the 'centroid' of project activity over time, it is important to remember that a complex system will involve many (perhaps thousands of) people whose activities may be spread over a large volume of the model.

Operationally, software systems engineering will encompass an egg-shaped solid centred on the Z-axis of this model. On a linear scale, this volume of activities will be centred rather high up on the Z-axis as, once a system is fully operational, the involvement decreases rapidly (except when major modifications are needed). Likewise, the extent of the activities in the X-Y plane is not regular, reaching out during certain activities to include other expertise and contracting in other places. Overall, however, the extent of activities carried out by the software systems engineer is limited at the start, expands to a maximum during design and construction of the actual system, and tapers off once the system is fully operational and modifications have either ceased or decreased to a very low level.

Source: Andriole and Freeman (1993)

SAQ 14

The aim of Andriole and Freeman is to plug what they see as a gap between the generic systems engineering model illustrated in [Figure 41](#) and the specific software engineering methodology shown in [Figure 40](#). Do you consider that the model that they present ([Figure 43](#)) achieves this aim?

Answer

While the logic of Andriole and Freeman's argument is clear, the conceptual gap between systems and software engineering is certainly a growing problem. The solution that they present has a number of flaws.

It is true that some generic activities such as goal setting, planning, design, and construction may have to be carried out at different stages of the project, but the model does not convey clearly how this operates. Do the authors intend that each generic activity is undertaken at each stage of the life cycle? Is what they intend close to the model shown in [Figure 44](#)?

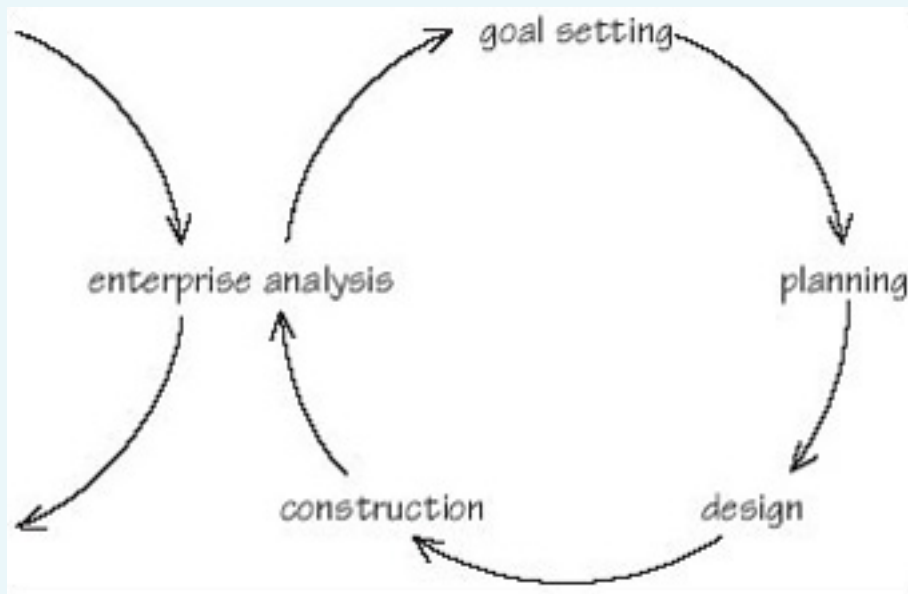


Figure 44 Andriole and Freeman's model as nested activities

It is intended as an operational basis for software systems projects but falls far short of the requirements for such a methodology. It does not provide a stable basis for project planning since the number of 'iterations' within the solid are not specified. To the wary eye the model looks less sure-footed than either the systems or the software engineering models that it hopes to replace.

4.6 Systems engineering: the recent development of a discipline

The recent development of systems engineering can be dated from two events. First, following the lead of the US Department of Defense and its introduction of standards for contractors, systems engineering was taken up by companies such as Boeing, Lockheed Martin and, in the UK, British Aerospace, Marconi and the Lucas Group. Second, in August 1990, a group of individuals interested in systems engineering met in Seattle (Box 10 – extract from a paper presented at the International Committee for Systems Engineering Symposium in 1996). This was the beginning of first the National Council on Systems Engineering and then the International Council on Systems Engineering.

Box 10 The recent development of systems engineering

In August of 1990 a group of 35 individuals, recognized for their interest, knowledge, and expertise in systems engineering, met in Seattle to discuss the need for a professional organization to foster the definition and understanding of systems engineering. The cost of attendance was submission of the individual's concept or definition of systems engineering. The submitted definitions were categorized into four groups. The first group believed systems engineering to be a functional discipline of system engineers with the practice restricted to technical activities. The second group considered systems engineering to be a discipline of system engineers with practice including both technical and management activities. The third group expressed systems engineering as a set of technical activities practiced by any required discipline, not just system engineers. And, the fourth group believed systems engineering to be a set of technical and management activities practiced by any required discipline.

Thus, from the beginning, INCOSE has been composed of persons with differing views of what systems engineering is. It is because of these different views that INCOSE did not attempt to define the term. Doing so was considered as being counter productive to the purpose and objectives of the new Council. The intent of the founders was to include all who shared the vision that systems could be developed and produced at a lower cost, delivered on schedule, and provided with expected performance attributes if more system qualified engineers and better processes, methods and automated tools were available.

Recognition of systems engineering

In the years since inception of INCOSE, enterprises have been found in which system engineers, systems engineering organizations, and the term systems engineering itself are not recognized as needed disciplines, organizations, or practices. To such enterprises, systems engineering is analogous to high overhead, too much unnecessary documentation, and other costly Department of Defense oversight practices. This is not to say that these enterprises do not accomplish systems engineering. They do, and in many enterprises with relative success. They could, however, like most enterprises with system engineers, systems engineering organizations and established systems engineering practices, accomplish the engineering of systems more efficiently and effectively if a disciplined, comprehensive systems engineering approach or process were utilized.

Enterprises that have shunned systems engineering because of perceived negative connotations, have adopted and fostered terms such as Concurrent Engineering and Integrated Product (and Process) Development. The trend lately is for enterprises with a history of embracing systems engineering, including the Department of Defense, to adopt these new terms to avoid the confusion related to systems engineering. Much of this can be directly attributed to a lack of appreciation of exactly what systems engineering is.

Commercial systems engineering standards

This confusion and the perceived need for other terms to describe the engineering of systems is unfortunate, but understandable. The animosity (often hostility) toward systems engineering is one reason new commercial systems engineering standards (EIA/IS 632 and IEEE 1220 – 1994) have not received wide acceptance, even within INCOSE.

Whereas system engineers look at these standards as much broader than what they do, non-system engineers look at the standard titles and come up with the

conclusion that the standards are not relevant to their work; that they are only applicable to system engineers. The adage 'don't judge a book by its cover' is applicable in this case.

The intent of these two new standards is well denned in their respective forward and scope sections. The purpose of the EIA standard is to improve the engineering of systems in oversight or contractual instances. The IEEE standard purpose 'is to provide a standard for managing a system from initial concept through development, operations, and disposal.' It explains what any enterprise must do to engineer a system. Its focus is on commercial, non-oversight developments.

Neither standard restricts its tasks to what a system engineer does or for which a systems engineering organization is responsible. Specifically, teams and teamwork are called out to include personnel to ensure that quality factors related to predictability, testability/verifiability, deployability, operability, supportability, trainability, and disposability are designed into system products. Additionally, both standards call for inclusion, as appropriate, of customers/users, subcontractors, and other non-engineering personnel such as marketing, legal and contracting on interdisciplinary teams.

Two processes important to engineering a system are the central focus of the IEEE systems engineering standard. These are the life cycle development process and the systems engineering process. The systems engineering process is the engine, recursively applied, that drives the evolution and maturity of the system through successive stages of development.

Systems engineering in relation to concurrent engineering and integrated process and product development

When the systems engineering envisioned in the standards is compared with the explained concepts and scope of Concurrent Engineering and Integrated Product and Process Development one finds that the purpose and scope of life cycles tasks are essentially the same, that the focus on downstream specialities in upstream design activities is the same, and that the utilization of automated tools is the same. The essential difference, which could lead one to consider the systems engineering approach described in the IEEE and EIA standards as being more comprehensive, is the inclusion of a comprehensive systems engineering process utilized recursively to mature the system solutions, attain customer satisfaction, and satisfy organizational commitments and public expectations.

So, the problems associated with acceptance of systems engineering appear to be based on lack of understanding on just what systems engineering is and what its purpose is.

Source: Lake (1996)

The other modern development of a variant of systems engineering occurred in the UK from the late 1970s onwards. Under the direction of John Parnaby, Lucas Engineering and Systems Department was established to improve the effectiveness of the disparate group of factories that made up the group. Many were in need of urgent reorganisation and updating in terms of their manufacturing approaches and methods. In response to this need, Parnaby and his staff developed methodologies for product and manufacturing and business systems engineering. Parnaby (1995) stated:

A system is an integrated combination of components designed to follow a common purpose. A systems philosophy demands that an unco-ordinated, piecemeal activity is replaced by a co-ordinated approach in which the identity of the separate parts of the system are subsumed by the identity of the total system. Systems engineering is an art, based on control systems principles, for designing complex systems. Individual elements and subsystems are fitted together to achieve an overall system purpose objective in the most effective way, at the lowest cost, with minimum complexity.

The Lucas approach to manufacturing systems engineering (MSE) is illustrated in [Figure 45](#). This illustrates the activities undertaken as part of the concept design of the system. The methodology lays particular stress upon the need to establish business targets and market plans as the basis for the design of the system. The methodology is designed to be implemented through the work of a multidisciplinary taskforce. The members of the taskforce can be classified into the three groups shown in [Figure 46](#).

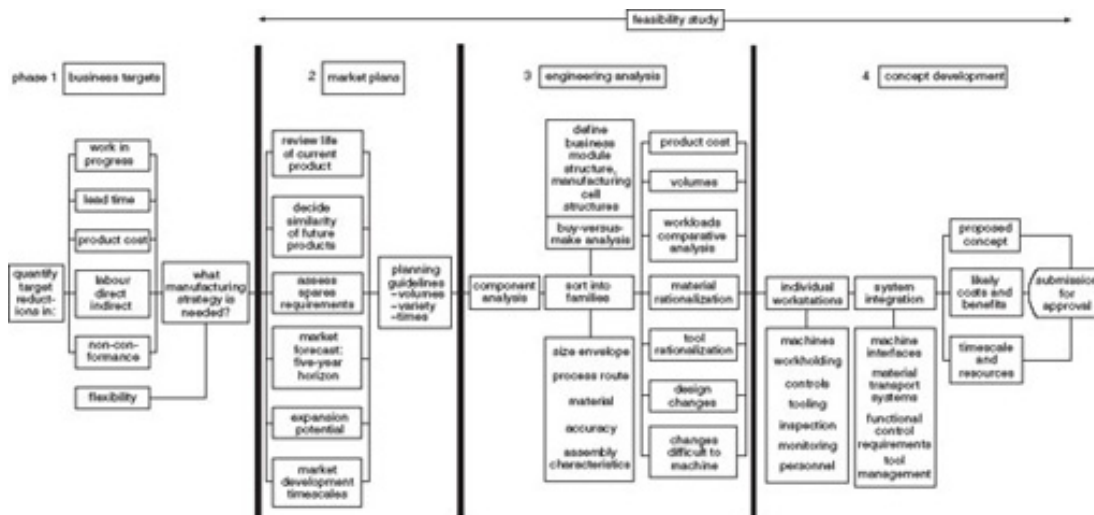


Figure 45 Manufacturing systems engineering methodology

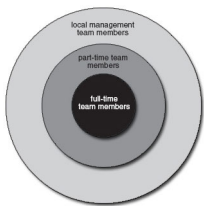


Figure 46 The composition of the MSE taskforce

As the statement made in the last paragraph of the extract in Box 10 suggests, there remains some uncertainty about the nature and extent of systems engineering. This uncertainty is reflected in the numerous definitions of systems engineering that are current. A selection of these definitions is given in Box 11.

Box 11 Definitions of systems engineering

US Military Standard MIL-STD-499A Systems Engineering:

‘... the application of scientific and engineering efforts to:

transform an operational need into a description of system performance parameters and a system configuration through the use of an iterative process of definition, synthesis, analysis, design, test, and evaluation;
 integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design;
 integrate reliability, maintainability, safety, survivability, human engineering, and other such factors into the total engineering effort to meet cost, schedule, supportability, and technical performance objectives.'

US Military Standard MU-STD-499B Systems Engineering:

'Systems engineering is an interdisciplinary approach to evolve and verify an integrated and optimally balanced set of product and process designs that satisfy user needs and provide information for management decision making.'

DSMC Systems Engineering Management Guide, *January 1990, pp. 1–2:*

'... is the management function which controls the total system development effort for the purpose of achieving an optimum balance of all systems elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize the overall system effectiveness.'

IBM Federal Systems Company definition used as the basis for divisions, internal SE practices and education:

'The iterative controlled process in which users' needs are understood and evolved, through incremental development of requirements specifications and system design, to an operational system.'

Joe DeFoe and Jim McAuley; November 1991:

'The processing of building real things to solve real problems within technological, environmental, economic, legal, ethical, cultural, and institutional constraints.'

Computer Aided Systems Engineering, *Howard Eisner, Prentice Hall, 1988, page 17:*

'... an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near-optimal manner, the full range of requirements for the system.'

Field Manual: System Engineering, *FM 770–78 Headquarters, Department of the Army, April 1979, pp. 1–2. Summary definition in the introduction of the manual:*

'The transforming of an operational need into a description of system performance parameters and a system configuration.'

Field Manual: System Engineering, *FM 770–78 Headquarters, Department of the Army, April 1979, p. C-2. More extensive definition found in the glossary:*

'The selective application of scientific and engineering efforts to:

1. transform an operational need into a description system configuration which best satisfies the operational need according the measures of effectiveness;
2. integrate related technical parameters and ensure compatibility of all physical, functional, and technical program interfaces in a manner which optimizes the total system definition and design;

3. integrate the efforts of all engineering disciplines and specialities into the total engineering effort.'

Fundamentals of Systems Engineering at NASA, INCOSE/ASEM Proceedings, October 1991, Robert G. Chamberlain and Robert Shishko, PhD, p. 23:

'... is a robust approach to the design and creation of systems to accomplish desired ends.'

P. M'Pherson, 'Systems engineering: A proposed definition', IEE Proceedings, vol. 133, pp. 330–331, Sep. 1986 (3):

'... is a hybrid methodology that combines policy analysis, design and management. It aims to ensure that a complex man-made system, selected from the range of options on offer, is the one most likely to satisfy the owner's objectives in the context of long-term future operational or market environments.'

INCOSE:

'Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:

Operations

Performance

Test

Manufacturing

Cost & Schedule

Training & Support

Disposal

Systems Engineering integrates all the disciplines and speciality groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.'

Lucas:

'At the core of manufacturing system design is a relatively multidisciplinary engineering function called "Manufacturing Systems Engineering" which tries to view the manufacturing as a whole. It has a broader view point than the traditional disciplines of "industrial engineering" and "production engineering".'

Source: INCOSE (1997)

SAQ 15

From the definitions of systems engineering in Box 11 summarise the characteristics of the approach.

.....

Answer

My definition of the characteristics of systems engineering is:

A holistic, balanced approach. Systems engineering treats the problem, its environment, and the solution as a whole, not as a set of disconnected parts. The system and its environment have to be considered as an entity. Trade-offs between requirements, costs and timescales are an essential part of the work.

Creativity and domain knowledge. Systems engineering demands intense creativity, particularly in the system requirements and architectural design processes. The role is far more than coordination and management of other people's work. Systems are made by people, and are the products of clever minds. Indeed, success or failure often turns on the ability of a single brilliant person. The systems engineer must also understand the design domain in some detail to make logical decisions. We need to immerse ourselves in (but not dictate) the detail of systems, and be sensitive to changes that might affect us. Committee-driven approaches will not work. At different stages we need visionaries, inventors, makers, negotiators, organisers and implementers acting as part of a team. Process and tools are essential, but we must allow room for creativity, and sometimes use intuition instead of apparently hard facts.

Coping with risk and change. Risk and human fallibility are constants in system development and operations. We have to predict the operating environment of the system we are constructing. The perfect design is beyond our reach, reason has its limitations, and life is not predictable. Yet the systems must still work reliably and safely. Continuous change is hard enough to handle and requires constant awareness of the business environment. Discontinuous change is even more difficult to survive. Although systems engineering is about prediction and direction, the future is only partially discernible. We cannot anticipate every problem, but we must build systems which can cope with change.

Managing complexity. Modern systems are large, intricate and difficult to develop. They need discipline, organisation, traceability, planning, and allocation of responsibilities to ensure efficient development. Systems engineering processes and tools help contain the problems which result from this complexity, in both product and development process.

Reusing experience. An intricate system like a car is built on a century's work by a million engineers. Few core concepts need be invented, and we must be humble in learning from the past. This requires sharing of what we know, in the form of best practice for products and processes. This should be based on reality more than dry theorising.

A user-driven approach. Users are the experts at using the system. We have to build systems which satisfy them, or risk product or even business failure. Requirements management is therefore a key activity for the systems engineer, pervading the whole life cycle.

Whole-life consideration. Systems have a natural cycle, from conception and birth, and then through useful life until disposal. They may live on, transformed through evolution, or spawn successor products, reusing parts of the original blueprints. They are often coupled to other systems with which they must co-evolve to survive. Early decisions may impact much later in a product life cycle, and systems engineering must be aware of the downstream consequences.

Comprehension of multiple disciplines. Systems engineers inevitably interact with many specialist disciplines. They must understand enough about these specialist

skills to know what is vital for success, and how to deploy them. Although systems engineers are generalists, they must never imagine that they can do what they want or succeed on their own. Systems engineering is a service to others, providing a realistic framework for specialist knowledge.

Improving the performance of the team. Systems engineering involves many different roles through the life cycle. Developing a complex product is always a team effort, and methodologies must always take into account the interactions within a creative team. We must improve the teamwork to conquer difficult problems that no single person could have solved alone. The customer and the supplier must share risks from an early point, or the customer needs to take the systems engineering work to a point where risk is acceptable. The last few years have seen an increase in integrated stakeholder teams and concurrent engineering, based on partnership and information sharing.

Management of information. Systems engineering is not a technical discipline like software or mechanical engineering because it defines the products to be implemented by others. Information modelling and management are essential to direct the system development.

Linkage to business goals. Engineering of complex systems can succeed only within businesses providing trained staff, finances, facilities, practices and tools. Market-leading systems businesses commit to continuous system improvement, and spread these concepts through their supply chain. Systems engineering principles also apply to the building of those organisational support systems.

4.7 Summary

This topic has examined the historical development of systems engineering and modern concepts of the subject. It has discussed:

- the beginnings and early development of the subject as policy analysis
- the use of systems engineering in organisations
- the development of methodologies associated with information technology.

5 The original course team's approach to systems engineering

5.1 Introduction: the general framework

The general framework of systems engineering adopted in the course consists of: a hierarchy of elements; aims associated within its outputs and process; a set of principles; a division into technical and managerial components of the process.

The lexicon of system engineering used in the course contains the hierarchy of elements:

- *strategy*: meaning the accumulated decisions concerning the areas in which an organisation operates and its longer-term aims
- *programme*: this may be either a single, very large contract or a number of projects that are linked by a business, technological or other logic
- *project*: a set of activities required to produce a specified set of outputs
- *work package*: a subset of the activities within a project that have a disciplinary (functional), technological or other logic
- *task*: a specific activity.

These terms and the levels of activity to which they relate form the hierarchy shown in [Figure 47](#). The relationship of this to the systems structure discussed in the Section 4, and to enabling and operating environment systems, is also indicated. The strategy level of activity is associated with the 'Enterprise Environment Management Process' identified in the ISO/IEC 15288 standard.

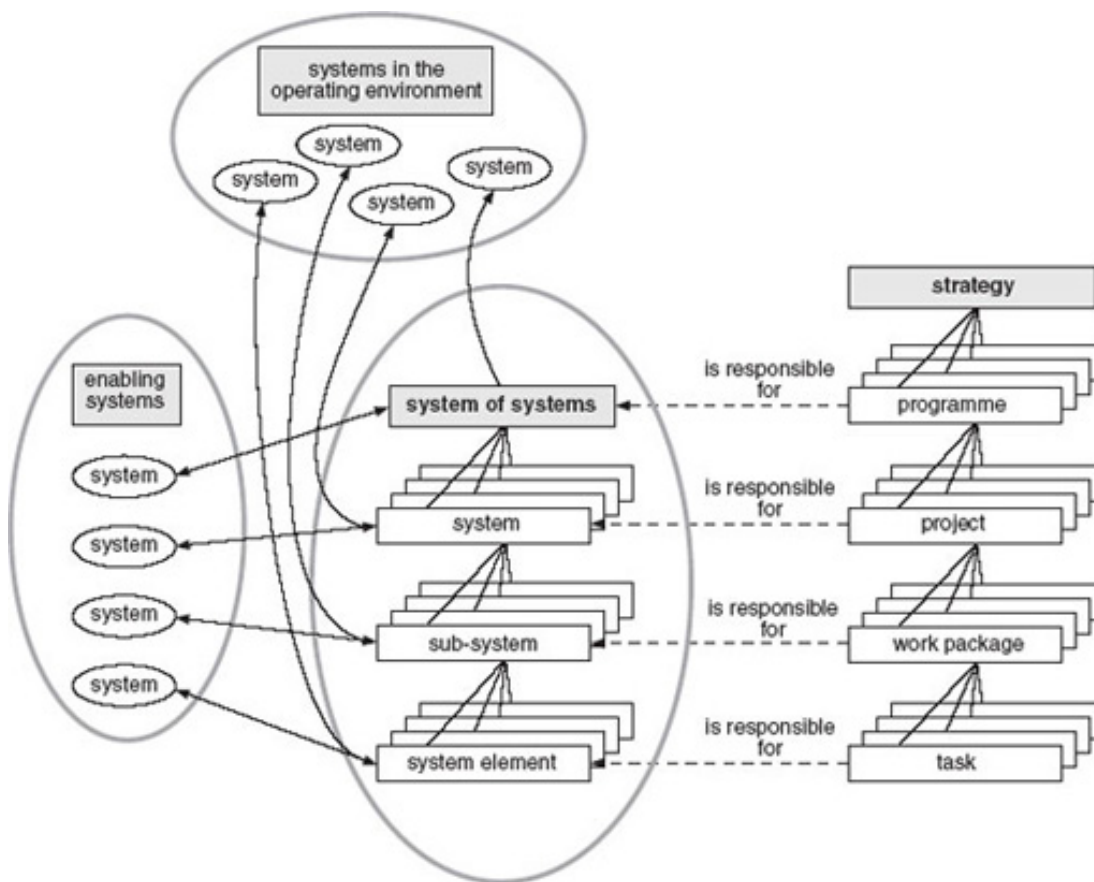


Figure 47 Structure, decomposition and relationships in systems engineering

Various definitions of systems engineering and its characteristics were discussed in Section 2. The definition adopted by the course team is:

Systems engineering is a set of principles, methods and techniques applied to all the tasks involved in all the life cycle stages of a complex system.

5.2 The aims and principles of system engineering

The *aims* of systems engineering can be divided into those to do with its outputs and those associated with the process itself. As far as its *outputs* are concerned, systems engineering aims to ensure that:

- the requirements of all the stakeholders are taken into account in engineering the system
- the system, as engineered and realised, meets the requirements of stakeholders
- the system, while meeting the requirements of stakeholders, should be designed and implemented in such a way as to minimise its negative impact on society and the physical environment.

Systems engineering as a *process* aims to:

- achieve the realisation of the system within time and cost targets
- ensure that the realisation of the system is profitable for a business enterprise or effective in a not-for-profit organisation

- ensure that any risks associated with the realisation of the system are minimised
- provide a framework in which the combination of all those associated with the realisation of the system is maximised and their professional development furthered
- provide a process whereby the individual and team learning that takes place during the realisation of the system becomes part of the intellectual capital of the organisation.

There are seven *principles* of systems engineering.

1. *Systemicity*: a systems viewpoint is adopted, including, in particular, taking a holistic, balanced approach.
2. *Betterment* (from definitions of engineering): whatever is done should contribute to the amelioration of the human condition and improve or minimise damage to the sentient and non-sentient environment.
3. *Relevance*: systems engineering effort is directed towards achieving the wider strategic aims of the organisation.
4. *Inclusivity*: the requirements of all stakeholders in the system are taken into account in its realisation. This follows from the concept of holism.
5. *Comprehensiveness*: all the stages of the system's life cycle are addressed.
6. *Multidisciplinary*: since systems engineering is concerned with the realisation of complex systems, a balanced contribution from different, appropriate specialist disciplines is required.
7. *Personal and team growth and development*: systems engineering contributes to the growth of knowledge within the team and organisation.

The aims and principles of systems engineering provide the basis for decision and action. The final element in the general framework used in the course is a division between the technical and managerial components (Sage, 1981; Sage, 1994; Shenhar, 1999). Shenhar represents these two elements with the diagram shown in [Figure 48](#).

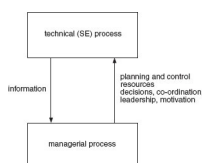


Figure 48 Shenhar's model of technical and managerial components of systems engineering. Source: Shenhar, 1999, p. 115

Question 5

Look at Shenhar's diagram ([Figure 48](#)). What are the main criticisms that might be levelled at it?

.....

Answer

The main criticism that I would make of Shenhar's model is that it seems to imply a separation of the technical and managerial processes. Such a separation is undesirable and may be disastrous. A second objection is that Shenhar's model seems to imply that, while the technical process is specific to systems engineering, the managerial process is generic in character. This criticism may arise from the way that the diagram has been drawn. The text that accompanies the diagram

suggests the following as the most common activities of systems engineering managers.

1. The identification of an operational need with an opportunity to create a system to respond to this need.
2. Setting the exact system and functional requirements to ensure the best fit to customer needs.
3. Dividing and allocating the functional requirements into different subfunctions and modes of operation.
4. Choosing the system concept that will best fit these requirements.
5. Designing the system architecture, based on the chosen concept.
6. Dividing the system into separate sub-systems and components to ensure overall optimisation, fewest interfaces, and fewest mutual effects of the various sub-systems.
7. Optimising the specifications of the various sub-systems through simulation, analysis and trade studies.
8. Managing the interaction with various engineering groups that performed the design of the sub-systems while integrating various people and disciplines.
9. Performing the integration of the various sub-systems into a total system.
10. Evaluating the performance and qualifications of the final system through simulation and testing activities.
11. Demonstrating the operating system to customers and convincing them that it responds to their needs.

The following are the technical activities associated with systems engineering (Shenhar, 1999, pp. 116–18).

1. Need identification and customer linkage. To identify the need and the system opportunity by matching need and technical feasibility and be the link bond between customer needs and system idea and design during the entire process of system creation.
2. Requirements management. To develop a set of system and functional requirements based on customer needs.
3. Architecture and system design. To be the lead person in envisioning the system's concept, and to create the link between the system's requirements and the system's configuration.
4. Integration. To see the entire picture and how each part contributes to the performance and feasibility of the system as a whole. Also, to coordinate the work of the various disciplines and professions involved and manage the interfaces among them such that the result is an overall optimal system.
5. Analysis, development and testing. To collect data from various sources, perform modelling and simulation and analyse them as a basis for decision making to confirm that the system is designed to its requirements; and to test and verify that the system built will meet these requirements as designed.
6. Process management. To plan, document, direct and follow the systems engineering process.
7. Technical and risk management. The process of systems engineering involves technical and trade-off decisions and the resolution of technical problems and conflicts at different interface points. These conflicts are primarily professional rather than personal, and reflect the different views of the distinct disciplines involved in creation of the system. This role also involves risk assessment on

various system elements and overall risk management during the system creation.

8. Leading, coordinating and managing. In addition to being a technical manager, the systems engineer must be a manager of activities and leader and coordinator of people. The job includes dealing with work plans, schedules, and budgets, but also working with people – organising their work, motivating them, communicating with them, and dealing with their needs.
9. Logistics and operations management. To consider and include maintenance, operation, logistics, and disposal concerns during the requirements, design, and development phases, and to 'escort' the users during the operational phase of the system, to 'break them in', to answer questions and solve anomalies.
10. Information management. To see the overall information needs of the system, plan the forms and means in which information will be created, disseminated and stored, and direct the process of information sharing and configuration control.

The technical and the managerial have to be seen within the context of the whole that is systems engineering. Its two components rely symbiotically on each other and both are systems engineering specific. The inextricably linked relationship of the technical and managerial are shown at the centre of [Figure 49](#), which also indicates that the two components are special subsets of the wider general subjects of engineering and management.

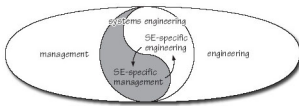


Figure 49 The yin and yang of systems engineering

The INCOSE model of the technical component of systems engineering is shown in [Figure 50](#).

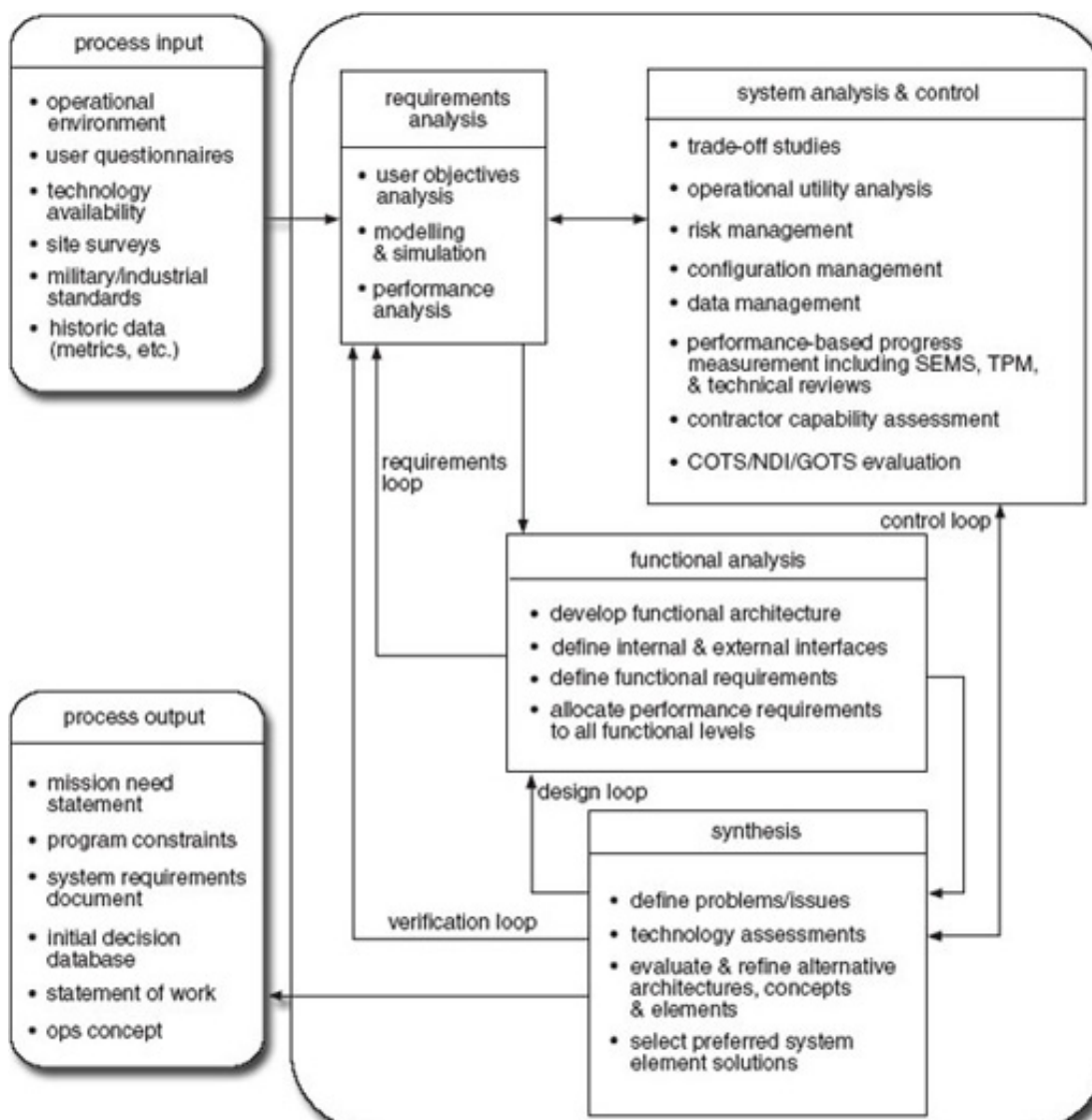


Figure 50 The INCOSE model of the technical components of systems engineering

5.3 The systems engineering methodology used in the course

The aim of systems engineering is to achieve a solution that is effective and sustainable through its life cycle, together with the associated processes and facilities needed to realise the system and introduce it into the real world. Therefore it is important that systems engineering is itself conducted in full consideration of the following five systems:

- the technology development system that provides new or modified technology for the other systems
- the product or service system – this is the primary output of the transformation process
- the realisation system – the resources needed to produce or effect the product or service system
- the sales and marketing system – most changes to systems have to be sold in some way or another to purchasers or users

- the support system – most systems have a life of several years, during which time they need to be supported and maintained. The forms that these requirements may take are technical support, spares and maintenance, training, logistics, and the possibly important issues surrounding disposal after use.

Any one or, more rarely, more than one of these systems can be the system of interest – that is, the system that is at the focus of the systems engineering effort. For example, the development of an aircraft would imply a new or enhanced product system. In a systems engineering project to design and implement a new production plant the realisation system would be the primary focus.

A change to any one of the five systems will almost invariably result in changes to the others – the secondary systems. Assessing the degree of potential change in the overall system of systems is an important activity that takes place during the tendering or feasibility process.

Core techniques for assessing the degree of change in the system of systems are illustrated by the systems engineering difference assessment web (SEDAW). This is a single representation on a polar graph of the amount of change involved in each of the system of systems. For example, a major new model programme in the car industry involved changes in the other four systems. The Programme Development Team assessed the degree of change by carefully evaluating how much each system would have to be changed to accommodate the new model. The degree of difference in each system is shown in [Figure 51](#).



Figure 51: The SEDA

It is important when making such an assessment to distinguish between doing a different thing and doing something differently. For example, at one extreme the fashion industry changes its product lines regularly but with hardly a difference to its systems. The other

end of the scale would be represented by a new airliner requiring significant new technologies to be incorporated into the product and the production systems.

Any system will have what is termed a life cycle, and ISO/IEC 15288 defines the stages in this as:

- concept
- development
- production
- utilisation
- support
- retirement.

Associated with these stages are the set of technical processes:

- stakeholder requirements definition
- requirements analysis
- architectural design
- implementation
- integration
- verification
- transition
- validation
- operation
- maintenance
- disposal.

Conclusion

This course has covered the background to systems engineering. It began by addressing the question 'Why is systems engineering important?' Two reasons were discussed:

- projects go wrong, and the increasing incorporation of software means that they go wrong more often now than in the past
- complication, complexity and risk are all increasing and need to be managed.

In the second section I examined the development of engineering and concluded that, until recently, it has been methodologically naive and has lacked the awareness of the importance of process shown in science. The systems approaches were then discussed in Section 3 in terms of:

- a set of concepts
- two generic methodologies
- analytical and design techniques.

Section 4 addressed the question 'What is systems engineering?' by looking at its origins and development during the second half of the twentieth century. It covered:

- the beginnings and early development of the subject as policy analysis
- the use of systems engineering in organisations
- the methodologies that developed in association with information technology
- recent developments in systems engineering.

Finally, in Section 5 I explained the original course team's approach to systems engineering in terms of:

- a hierarchy of elements from business strategy to specific activities or tasks
- a set of aims to do with the outputs and process of systems engineering
- a methodology consisting of a set of life cycle stages and technical processes.

Keep on learning



Study another free course

There are more than **800 courses on OpenLearn** for you to choose from on a range of subjects.

Find out more about all our [free courses](#).

Take your studies further

Find out more about studying with The Open University by [visiting our online prospectus](#).

If you are new to university study, you may be interested in our [Access Courses](#) or [Certificates](#).

What's new from OpenLearn?

[Sign up to our newsletter](#) or view a sample.

For reference, full URLs to pages listed above:

OpenLearn – www.open.edu/openlearn/free-courses

Visiting our online prospectus – www.open.ac.uk/courses

Access Courses – www.open.ac.uk/courses/do-it/access

Certificates – www.open.ac.uk/courses/certificates-he

Newsletter –

www.open.edu/openlearn/about-openlearn/subscribe-the-openlearn-newsletter

References

Abram, J. (2001) *The Contribution of the Product Definition Process to a Successful High Volume Software Application*, unpublished MSc dissertation, Milton Keynes, The Open University, p. 48.

Andriole, S.J. and Freeman, P.A. (1993) 'Software systems engineering: the case for a new discipline', *Software Engineering Journal*, vol. 8, no. 3, pp. 165–79. Reprinted in Dorfman and Thayer (1997), pp. 29–43.

- Anscombe, E. and Geach, P.T. (selectors and eds) (1970) *Descartes' Philosophical Writings*, London, Thomas Nelson.
- [Arup \(2000a\)](#) (Accessed 21 August 2007).
- [Arup\(2000b\)](#) (Accessed 4 December 2000 – no longer accessible).
- Arup (2005) '[The Millennium Bridge](#)' [(Accessed 21 August 2007).
- [Autodesk \(2000\)](#) Home page, –123112,00.html (Accessed 21 August 2007).
- Basalla, G. (1988) *The Evolution of Technology*, Cambridge, Cambridge University Press.
- Bauer, F.L (1993) Software engineering – a European perspective (Foreword) in Dorfman and Thayer (1997), p. 75.
- Boehm, B.W. (1988) 'A spiral model of software development', *IEEE Computer Magazine*, vol. 21, no. 5, pp. 61–73.
- ISO/IEC 15288 (2002(E)) BSI, 389 Chiswick Road, London W4 4AL.
- Brooks, F.P. (1987) 'No silver bullet. Essence and accidents of software engineering', *Computer*, vol. 20, no. 4, pp. 10–19. Reprinted in Dorfman, M. and Thayer, R.H. (1997), pp. 13–22.
- Cardwell, D. (1994) *The Fontana History of Technology*, London, Fontana Press.
- Checkland, P.C. (1993) *Systems Thinking: Systems Practice*, Chichester, UK, Wiley.
- Checkland, P. and Scholes, J. (1990) *Soft Systems Methodology in Action*, New York, Wiley.
- [Citizens Advice Bureau \(2005\)](#) 'Working tax credit' .htm#working_tax_credit (Accessed 21 August 2007).
- de Neufville, R. and Stafford, J.H. (1974) *Systems Analysis for Engineers and Managers*, McGraw-Hill, London.
- Dewey, J. (1933) *How We Think*, Boston, Mass., D.C. Heath and Company.
- Dorfman, M. and Thayer, R.H. (eds) (1997) *Software Engineering*, Los Alamitos, California, IEEE Computer Society Press.
- Eisner, H. (1988) *Computer-aided Systems Engineering*, Englewood Cliffs, New Jersey, Prentice-Hall.
- Emery, F.E. (1981) 'The assembly line: its logic and our future' in Emery, F.E. (ed.) (1981) *Systems Thinking*, vol. 2, Harmondsworth, Penguin Books.
- Emery, F.E. and Trist, E.L. (1960) 'Socio-technical systems' in Churchman, C.W. and Verhulst, M. (eds) *Management Science, Models and Techniques*, vol. 2, Oxford, Pergamon Press, pp. 83–97. Reprinted in Emery, F.E. (ed.) (1969) *Systems Thinking*, Harmondsworth, Middlesex, Penguin, pp. 322–37.
- Engstrom, E.W. (1957) 'System engineering – a growing concept', *Electrical Engineering*, vol. 76, pp. 113–16. Quoted in Hall (1962), p. 13.
- EPC (2000) *The EPC Engineering Graduate Output Standard: Interim Report of the EPC Output Standards Project*, Coventry, Engineering Professors' Council.
- Faulk, S.R. (1997) 'Software requirements: a tutorial', in Dorfman and Thayer (1997), pp. 82–103.
- Ford, H. (1922) *My Life and Work*, London, Heinemann.
- Foucault, M. (1974) *The Order of Things. An Archaeology of the Human Sciences*. London, Tavistock Publications.
- Hall, A.D. (1962) *A Methodology for Systems Engineering*, New York, Van Nostrand.
- Hoos, I.R. (1972) *Systems Analysis in Public Policy: A Critique*, Berkeley, University of California Press.

- [INCOSE \(1997\)](#) 'A systems engineering partially annotated bibliography and resources list' [online] (Accessed October 2002 – no longer accessible).
- Kaposi, A. and Myers, M. (1994) *Systems, Models and Measures*, London, Springer-Verlag.
- Kerola, P. and Freeman, P.A. (1981) 'Comparison of life-cycle models', *Proceedings of the 5th Conference on Software Engineering*, Los Alamitos, California, IEEE Computer Society, pp. 90–9.
- Kranzberg, M. and Davenport, W.H. (1972) *Technology and Culture. An Anthology*, New York, Schicken Books.
- Lake, J.G. (1996) 'Unravelling the systems engineering lexicon', paper presented at the *1996 International Committee for Systems Engineering (INCOSE) Symposium*. Reprinted in Martin (1997).
- Lilienfeld, R. (1978) *The Rise of Systems Theory: An Ideological Analysis*, New York, Wiley.
- Locke, J. (1997 [1706]) *An Essay Concerning Human Understanding*, (ed. R. Woolhouse), London, Penguin Books.
- Macauley, L.A. (1996) *Requirements Engineering*, London, Springer-Verlag, pp. 30–3.
- Maguire, J.R. and Wyatt, T.A. (2002) *Dynamics Design and Practice Guide*, 2nd edn, London, Institution of Civil Engineers.
- Martin, J.N. (1997) *Systems Engineering Guidebook: A Process for Developing Systems and Products*, Boca Raton, Florida, CRC Press, pp. 235–48.
- OED (1989) *Oxford English Dictionary*, 2nd edn, Volume 4, Oxford, Clarendon Press.
- Ormerod, P. (1998) *Butterfly Economics. A New General Theory of Social and Economic Behaviours*, London, Faber and Faber, pp. 1–10.
- Parnaby, J. (1995) Inaugural address to the Institution of Electrical Engineers, 5 October, London.
- Poincare, H. (1995 [1903]) *Science and Method* (quoted in Gell-Martin, M. (1995) *The Quark and the Jaguar. Adventures in the Simplex and the Complex*, London, Abacus, p. 25).
- Pugh, S. (1991) *Total Design. Integrated Methods for Successful Product Engineering*, Harlow, England, Addison-Wesley.
- Quade, E.S. (ed.) (1967) Introduction to *Analysis for Military Decisions*, Chicago, Rand McNally, p. 3. Quoted in Hoos (1972) p. 44.
- Rae, J.B. (1960) 'The "know-how" tradition: technology in American history', *Technology and Culture*, vol. 1, 2, pp. 139–50.
- [RAND \(2000\)](#) 'About RAND' [online] (Accessed 21 August 2007).
- Rolt, L.T.C. (1970) *Isambard Kingdom Brunel*, London, Penguin Books.
- Royal Academy of Engineering (2000) 'The universe of engineering. A UK perspective', a Report prepared by a joint Royal Academy of Engineering/ Engineering Council Working Group under the chairmanship of Sir Robert Malpas CBE FREng., London, The Royal Academy of Engineering, London.
- Royal Ordnance (1997) *Systems Engineering in Royal Ordnance*, p. 20.
- Sage, A.P. (1981) 'Systems engineering: fundamental limits and future prospects', *Proceedings IEEE*, vol. 69, pp. 158–66.
- Sage, A.P. (1994) 'The many faces of systems engineering', *Systems Engineering*, vol. 1, pp. 43–60.
- Sage, A.P. (2000) 'Systems engineering education', *IEEE Transactions on Systems, Man and Cybernetics – Part C: Application and Reviews*, vol. 30, pp. 164–74.

- Sage, A.P. and Palmer, J.D. (1989) *Software Systems Engineering*, New York, Wiley.
- Sage, A.P. and Rouse, W.B. (eds) (1999) 'An introduction to systems engineering and systems management' in *Handbook of Systems Engineering and Management*, New York, John Wiley & Sons, pp. 1–58.
- Schofield, R.E. (1963) *The Lunar Society of Birmingham. A Social History of Provincial Science and Industry in Eighteenth Century England*, Oxford, The Clarendon Press, p. 3.
- Shelley, M. (1992 [1818]) *Frankenstein, or the Modern Prometheus* (ed. M. Hindle), London, Penguin Classics.
- Shenhav, A.J. (1999) 'Systems engineering management: the multi-disciplinary discipline' in Sage, A.P. and Rouse, W.B. (1999), pp. 114–36.
- Sloan, A.P. Jr. (1965) *My Years with General Motors*, London, Pan Books.
- Smith, A. (1976 [1776]) *An Inquiry into the Nature and Causes of the Wealth of Nations* (ed. E. Cannan), Chicago, University of Chicago Press.
- Stevens, R., Brook, P., Jackson, K. and Arnold, S. (1998) *Systems Engineering. Coping with Complexity*. London, Prentice Hall Europe.
- [Transport for London \(2005\)](#) 'Congestion charging – the 11 key numbers' [online] (Accessed July 2005 – no longer accessible).
- von Bertalanffy, L. (1968) *General System Theory*, New York, George Braziller.
- Whewell, W. (1968 [1858]) *Theory of Scientific Method* (ed. R. Butts), Indianapolis/Cambridge, Hackett Publishing Company.
- Winner, L. (1977) *Autonomous Technology. Technics-out-of-Control as a Theme in Political Thought*, Cambridge, Mass., The MIT Press.
- [Yahoo \(2000\)](#) (Accessed 4 December 2000 – no longer accessible).

Acknowledgements

The following material is Proprietary and is used under licence:

Course image: [The Cookiemonster](#) in Flickr made available under [Creative Commons Attribution-NonCommercial 2.0 Licence](#).

Various pages: [Arup](#), O., material accessed in January 2002 and December 2000, from.

Box 1: Inman, P. 'Chaotic scheme that left families relying on food parcels', *The Guardian*, 6 July 2005. © Guardian News and Media Ltd 2005.

Box 2: 'Fly-away drones put robot air force plans off course', *The Sunday Times*, 22 June 2003. News Syndication International. Graphic by Julian Osbaldstone and Ian Moores.

Box 3: Ormerod, P. (1998) *Butterfly Economics: A New General Theory of Social and Economic Behaviour*, Faber and Faber.

Box 5: The Universe of Engineering: A UK Perspective, June 2000, The Royal Academy of Engineering. © Sir Robert Malpas.

Box 9: Dorfman, M. and Thayer, R.H. (1997) *Software Engineering*, IEEE Computer Society Press. © Copyright 1997 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Box 10: Lake, J.G. (1996) 'Unravelling the systems engineering lexicon'. Paper presented at the 1996 International Committee for Systems Engineering Symposium.

Fire Department Deployment Analysis – Modelling in Public Policy

Based on Walker et al. (eds) *Fire Department Deployment Analysis: A Public Policy Analysis Case Study*. The Rand Fire Project, Elsevier North Holland Inc. The Rand Corporation.

The Roskill Commission – Using Cost–Benefit Analysis for Public Policy

Hall, P. (1980) 'London's Third Airport', *Great Planning Disasters*, Penguin Books Ltd.

Figure 1: Courtesy of Autodesk, Farnborough.

Figures 2, 40, 41 and 43: Dorfman, M. and Thayer, R. H. (1997) *Software Engineering*, IEEE Computer Society Press. © Copyright 1997 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Figure 17a, clockwise from top left:

© Mary Evans Picture Library,

© National Motor Museum, Beaulieu,

© National Motor Museum, Beaulieu,

© Ford Motor Company

© Mehau Kulyk/Science Photo Library,

© National Motor Museum, Beaulieu.

Figure 17b, clockwise from top left:

© Mary Evans Picture Library, courtesy of Christie's South Kensington,

© Mary Evans Picture Library,

© Françoise Sauze/Science Photo Library,

© courtesy of MP3-Plus Net,

© courtesy of Sharp Electronics (UK),

© P M Northwood,

© P M Northwood.

Figure 19: *The Universe of Engineering: A UK Perspective*, June 2000, The Royal Academy of Engineering. © Sir Robert Malpas.

Figure 20: Pugh, S. (1991) *Total Design Integrated Methods for Successful Product Engineering*, Addison-Wesley Publishing Company. Reprinted with permission from Pearson Education Ltd.

Figure 26: René Magritte, 'Ceci n'est pas une pipe', poster for a shop window, 1952, © DACS.

Figure 41: Eisner, H. (1988), *Computeraided Systems Engineering*, Prentice Hall.

Figure 42: Sage, A.P. and Rouse, W.B. (1999), *Handbook of Systems Engineering and Management*, John Wiley & Sons, Inc.

Figure 48: Shenhar, A. J. 'Systems engineering management: the multidisciplinary discipline', in Sage, A.P. and Rouse, W.B. (1999) *Handbook of Systems Engineering and Management*, John Wiley & Sons Inc.

Figure 50: INCOSE 1997 *A Systems Engineering Partially Annotated Bibliography and Resources List*, International Council on Systems Engineering.

Don't miss out:

If reading this text has inspired you to learn more, you may be interested in joining the millions of people who discover our free learning resources and qualifications by visiting The Open University - www.open.edu/openlearn/free-courses