SYSTEM LIFECYCLE

This lecture is on the “System Lifecycle”. Any engineering artefact goes through a "lifecycle" analogous to a living thing:

Birth - Growth - Adulthood - Death

In the case of an engineering product it is not really a cycle but a flow. However it is normally assumed that a replacement product will come along at the end of the products life (sort of like children) and everybody calls it the "System Lifecycle" so we will.

This sequence is also called a “waterfall” process which is perhaps a little better as an analogy.

There are also the “Procurement Cycle” which sometimes match the system life cycle and sometimes don’t – it depends on the product and customer supplier relation.

ASIDE – THE PRODUCT LIFECYCLE

There is also a term “Product Lifecycle” which is a waterfall description of a product’s life in the market place and is a concept for marketing and sales types. A typical description would be

1. New product development stage
very expensive - no sales revenue - losses

2. Market introduction stage
cost high - sales volume low - no/little competition - competitive manufacturers watch for acceptance/segment growth - losses

3. Growth stage
costs reduced due to economies of scale - sales volume increases significantly - profitability - public awareness - competition begins to increase with a few new players in the establishing market - prices to maximize market share

4. Mature stage
costs are very low as you are well established in market & no need for publicity. - sales volume peaks - increase in competitive offerings – prices tend to drop due to the proliferation of competing products - brand differentiation, feature diversification, as each player seeks to differentiate from competition with "how much product" is offered - very profitable

5. Decline or Stability stage
costs become counter-optimal - sales volume decline or stabilize – prices, profitability diminish - profit becomes more a challenge of production/distribution efficiency than increased sale

Unlike the Procurement Cycle this is NEVER related to the system’s lifecycle

THE CLASSIC SYSTEM LIFECYCLE

The engineering activities differ as various phases of the lifecycle are gone through. This includes the Systems Engineering activities (e.g. too late to do trade-offs after the product is built). Different systems engineering jobs are done in different phases to meet the objectives of that phase and all future phases of the lifecycle should be considered at every stage.

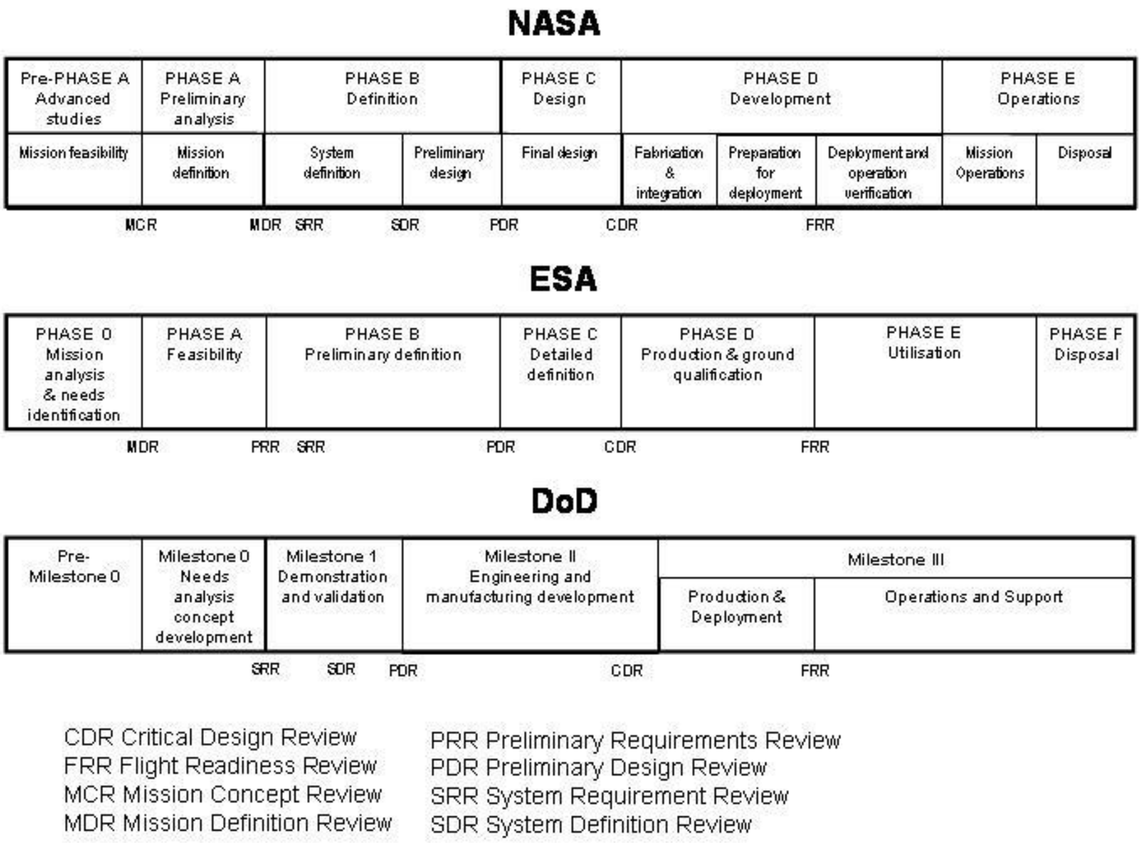
These lectures	**Wikipedia**	**Aslaksen and Belcher**
Concept Generation		
Requirement Generation	**Conceptual design**	**Definition**
System Level Design	**Preliminary system design**	**Analysis**
Design and Development	**Detail design and development**	**Design**
Production and Verification	**Production and construction**	**Implementation and Verification**
Utilisation	**Utilization and support**	**-**
Decommission	**Phase-out and disposal**	**-**

* Concept generation is before formal project funding and so is not included in many lifecycle definitions

Phases are often altered and defined to suit the various industries and products.

NASA/ESA PROJECT PHASES

The diagram below shows three lifecycles for spacecraft. NASA (National Aeronautics and Space Administration) and ESA (European Space Agency) both have defined project phases to take them through the system lifecycle. The USA DoD (Department of Defense) uses its normal procurement lifecycle.



Note that even with a single industry the definitions can change. ESA's version of the NASA cycles is slightly different – the result of slightly different philosophies. The most significant difference is that decommissioning is a separate Phase F rather than a sub-phase of E. The DoD procurement cycle is actually more suited to high production runs of aircraft or tanks and so does not fit quite so well with spacecraft when only 2 or 3 are going to be made.

Although there are many definitions of the system lifecycle I will tend to use the NASA terms in my lectures - partly because of my background and partly because it is the closest to the "classical" cycle. However it is very important you bear in mind other industries and countries use a) different terms and b) there are variants on the system lifecycle concept.

Warning people can also get it wrong. For example in "Spacecraft Systems Engineering" 2nd Edition by Fortescue and Stark the spacecraft lifecycle phases are given as

Phase A - Feasibility
Phase B - Detailed Definition
Phase C/D - Development, Manufacture, Integration and Test
Phase E - Launch Campaign
Phase F - Mission Operations.

which not only forgets disposal - a common mistake - but also has incorrect definitions for Phase E and F. This emphasises the point again. You will find lots of different terms, splits etc., even when you would think it should be the same. Sometimes the differences are genuine other times they are somebody getting it wrong. Go back to the defining procedure if in doubt.

Interestingly the Russian (soviet) space programme used a similar set of project phases.

1 - Project starts with "Initial Data" that acts a preliminary specification to act as a basis to start discussions with customer and contractor.

2 - These discussion lead to a "performance specification" which defines what the final product must do (the direct equivalent to the requirement specification). Interesting the final formalising of this specification after Contractors are involved mirrors the NASA / ESA practice.

3 - Preliminary design (as NASA / ESA).

4 - Technical Project (full development).

THE DOWNEY CYCLE

Another variation is the procurement lifecycle, known as the Downey Cycle, used by the British MOD after a report by Lord Downey in 1962. It is shown below: note especially:

- this is a procurement cycle that matches the system lifecycle
- the cost of each phase,
- inclusion of "Disposal" (normally forgotten),
- the different types of contracts,
- customer perspective on outputs and activities.

THE DOWNEY CYCLE

PHASE	Concept	Feasibility Study	Project Definition	Full Development	Production	In-Service	Disposal
ACTIVITY	Define operation objective	Identify risk	Assess and resolve risk	Detailed design and trials	Manufacture	Supply	
DEVELOPMENT COST %	0.5 - 5	5 -10	10 - 25	70-85	300	600	
CONTRACTS	Preliminary studies and demonstrators	Feasibility and project studies		Development and production		Repair spares	Sale
OUTPUTS	Reports and models	Reports and models		Satisfactory trials and acceptable equipment		Effective use	Profitable disposal

Since 1998 the MoD have used a new acquisition cycle called **CADMID** an acronym for the six stages involved - Concept, Assessment, Demonstration, Manufacture, In-Service and Disposal - which replaces the Downey cycle. However it is clear the basics are still the same.

If you look up "system lifecycle" on the web you will mostly find stuff related to computers and software – such as this example:

- **Project planning, feasibility study:** Establishes a high-level view of the intended project and determines its goals.
- **Systems analysis, requirements definition:** Refines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudocode and other documentation.
- **Implementation:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.
- **Maintenance:** What happens during the rest of the software's life: changes, correction, additions, moves to a different computing platform and more. This, the least glamorous and perhaps most important step of all, goes on seemingly forever.

But, as I hope I have already shown, the use of phased development approaches is much more widespread across all industries involved in system development.

And whatever the split the overall lifecycle logic always remains the same.

CONCEPT GENERATION

Concept generation is not a formal programme phase as it is the process by which a formal project starts. However as we have seen most organisations have a name for it

NASA = Pre-Phase A	ESA = Phase 0
DoD = Pre Milestone 0	Downey = Concept

All organisations involved in system procurement or development have teams looking at the future. Look out for departments with names like,

Future Studies,
Advanced Projects,
Business Development,

and similar titles. It is these departments that undertake the studies that lead to identification of new projects for the organisation.

Military and organised customers often have on going continuous assessments of their needs. In aerospace manufacturers normally have parallel departments often used to support the customer's future needs assessments. And all commercial organisations will be conducting market research even if only looking at their sales figures.

The classic driver for a new development project is to replace an existing product that is the end of its Product Lifecycle (decline). The manufacturer may wish to have a more competitive product to win back market share, or the customer may wish to replace worn out or obsolete systems.

Another reason to start a new development project is the identification of a promising new type of product (i.e. somebody has had a good idea!). This tends to be far more difficult to make happen. The average decision making executive has a close to zero ability to grasp what a good idea actually looks like. Also look at the first two stages of the Product Lifecycle – even if it going to succeed in the long run it is always a very bumpy ride before the new product is profitable and accepted. It takes a lot of bravery and determination to undertake it.

Either way the start of project should have as an input from the future projects group

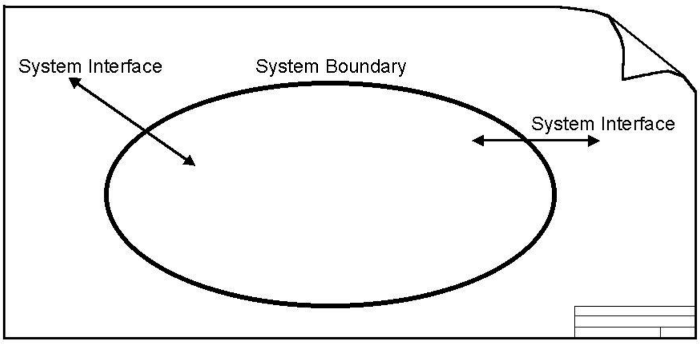
- A clear statement of what the project should achieve (the Purpose)

- Some preliminary work to establish feasibility to show it is worth spending serious money to explore further.

REQUIREMENT GENERATION PHASE

Requirement generation can be conducted either by the customer, or the producer with strong customer involvement, or producer with market analysis as a strong driver. Ultimately it serves to establish "what does the customer want the system to do?"

More abstractly the requirement generation phase produces a definition of what the system - is that is it defines the boundary and the system interfaces.



So what in detail is the output of the requirement generation phase?

Product Related Outputs:

- A requirement specification for the system
  This should match user needs.
- A demonstration that the system is feasible: technically and financially.
  Technically - can a system be made to meet the req. spec.?
  Financially - is the system affordable within budgetary constraints?

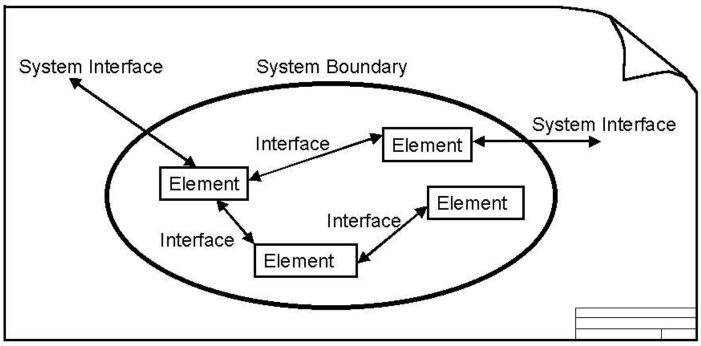
Management Related Outputs:

- Detailed plans and costs for next phase
  Normally includes bids from contractors to conduct the next phase
- Broader plans and costs for product lifecycle
  note that cost appears twice: for feasibility and for planning
- A decision to proceed from the funding authority
  If it is judged the customer wants (includes can afford) the system

This phase may finish with a formal meeting such as System Requirement Review (SRR) to make the decision to proceed. NASA and ESA both move the SRR a little into the next phase as by then the various contractors are involved and it gives a chance for them to make an input. Not quite a pure systems approach!

SYSTEM LEVEL DESIGN PHASE

In this phase contractors decide on the engineering approach they are going to offer the customer to meet the requirement specification. This is where the trade offs and optimisation work is conducted. In the end it produces a system level design – that is what the elements and interfaces within the system are going to be (in enough detail to write requirement specifications for them). So the system model now looks like this:



The key output document is the Design Specification (or equivalent documentation) which defines the system in reply to the Requirement Specification. Thus the Requirement Specification provides the boundary definition of the system and the Design Specification defines the element definition of the system.

Detailed (and accurate) costs of complete product lifecycle should be known: including enough information to make a commercial proposal on the price.

The system design may finish with a formal meeting such as the Preliminary Design Review (PDR). This is the point at which the final programme commitment is made as the cost and product are completely understood. If contractors have been competing for the development contract this is the point they are down-selected to one

THE EFFORT REQUIRED IN THE EARLY PHASES

Getting It Right!

These two phases lead to a decision to proceed with full project development (or not) and in theory once this approval has been obtained the project should get produced on time and to budget.

To ensure the decision is soundly based around 5% to 10% of the total development cost (i.e. all cost up to the point of production) should have be spent by the end of these two phases on the feasibility of the system (see numbers in Downey Cycle). That means a civil airliner project (typical development cost $10

Billion) will have spent $500 million or more. A modern fighter (typical cost $30 Billion) will have spent between $1 and $2 billion. These are large numbers to commit and it sometimes difficult to justify, but history shows if this sort of money has not been spent, and spent in a balanced way, technical and financial disaster is almost inevitable.

Experiments and Demonstrators

Part of where this money goes is in technology experiments and demonstrators systems. Most engineering industries have test projects of some sort to explore technical aspects of the design; for example the car industry has concept cars. The higher the technology level exploited the more such systems are important.

A practice that was once common when military aerospace systems were relatively cheaper, was to have contractors compete in the system design phase; often building demonstrators for "fly off" competitions. The USAF can still afford to do this sometimes as Lockheed YF22 Vs MacDonnell Douglas YF23 fly off shows. YF22 won leading to the F22 Raptor programme.

Lockheed YF22

McDonnell Douglas YF23

Lockheed F22 Raptor

More recently there was fly off between X-35 and X-36 leading to the Joint Strike Fighter (JSF) project - now the Lockheed Lightening II. Note they are misnamed an X plane was supposed to be a technology demonstrator, whereas these are design demonstrators and should have been designated YF-35 and YF-36.

There are three kinds of test system:

- **Experimental.** (or technology demonstrators) Used to explore and understand particular technologies in the pre-project or requirement generation phase

- **Demonstrators**: (or system design demonstrators) Used to demonstrate features of the proposed system in the system design phase

- **Prototypes:** (or pre-production models) Built to flight drawings to verify the complete design in detailed design phase (as part of qualification)

Example – EUROFIGHTER TYPHOON

Experimental
Proves a technology
Fly-by Wire Jaguar:
Proved fly by wire control of an unstable aircraft

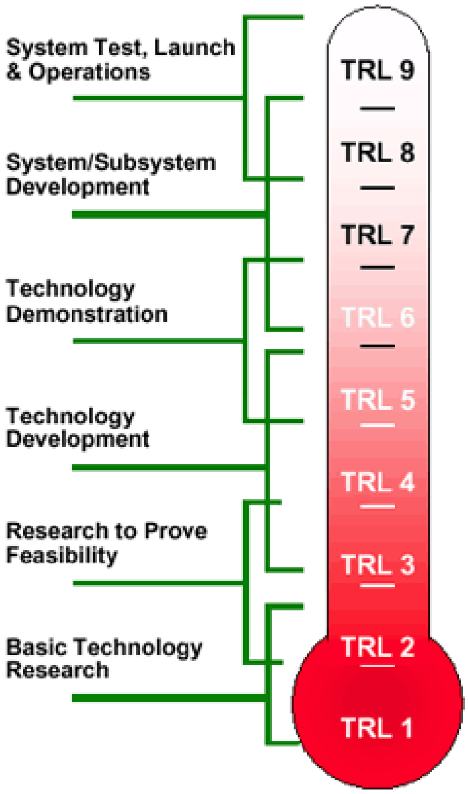
Design Demonstrator
Proves a system concept
Experimental Aircraft Programme (EAP). Proved CFRP structures and aerodynamics of Eurofighter.

Prototype
Proves the detailed design
Eurofigher Prototypes
The first Eurofighters used to prove design before full production starts.

The terms demonstrator and prototype are often misused. America used to have it very clearly defined: X-Planes were experimental, Y planes were design demonstrators, and prototypes were the pre-production test models. As the JSF programme shows (using X designation for system design demonstrator) they have rather lost the plot recently.

Make sure you use these terms carefully to demonstrate you understand the differences. In particular make sure you do not confuse prototype with demonstrator – this often done in the outside world, even by people you would have thought knew better, but shows everybody really understand that you haven’t a clue what is going on.

TECHNOLOGY READINESS LEVELS



Technology Readiness Levels (TRLs) were invented in 1974 by NASA and later formally adopted by US Air Force, European Space Agency, and the EU Commission. They are widely used today in general engineering.

It is a tool to assess the readiness of technology to be incorporated into a system. It is a waterfall process taking a new technology from wild vague idea to proven in operation.

It is good practice to undertake a Technology Readiness Review during the system design phase to ensure

a) the element requirement specifications only ask for the feasible

and

b) the risks and likely development effort are properly evaluated.

Because the later stages are demonstrations that the technology can be incorporated into a system there is often a confusion that this is part of a particular system lifecycle. It is not – it tells you the heritage of the technology you plan to incorporate. If the TRL is below 7 it tells you

a) have a risk

and

b) you may have to do some development work in that area.

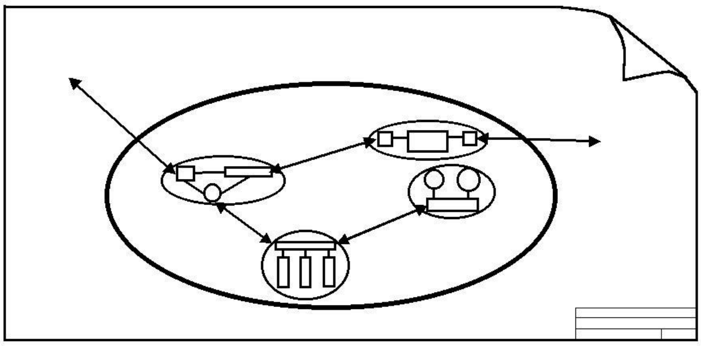
Even worse is the belief that a system has a TRL – it doesn't, only the technologies that make it up and have TRLs and they can be at many levels within a single system.

Another common mistake is to assume because a technology is at level 9 means there cannot be a problem with using it. Not so - remember it has been proven on another system (not yours) with a different requirement.

DETAILED DESIGN PHASE

Some text books imply systems engineering practically finishes at the system design phase and the issuing of all the element specifications. Not true! (Although it can get more boring as an academic concept). As detailed design proceeds system level monitoring ensures components meet specifications and that the system modelling and budgets are revised with the components actual parameters. System engineers also will need to assess the impact of components that do not meet their specification on the overall system specification.

The end result of the detailed design phase should be everything needed to start manufacture (drawings, tooling, etc.) and proof the system (as designed) meets the requirement specification. It turns the system conceptual design into a complete real design.

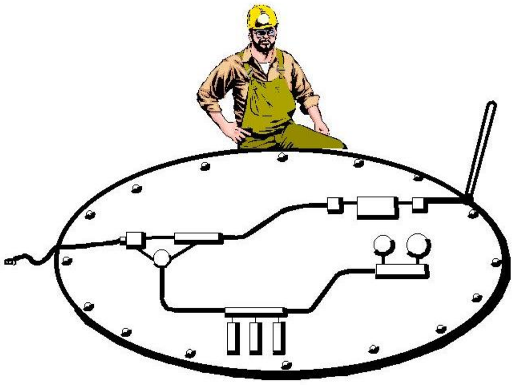


The detailed design phase finishes with a review such as the Critical Design Review (CDR), which approves the design for production. This means the system engineer must satisfy the paymaster that the design meets the requirements (i.e. it is qualified). The design is then frozen and alterations of any manufacturing drawing must undergo formal “Change Control”.

(Piece of advice: Although this phase is hardly glamorous for the systems engineer it is an excellent opportunity to learn the consequences of decisions made in earlier phases. It is essential experience. Engineers who have only worked on concept generation and requirement generation are a menace and unfortunately, because full development projects are now so rare, there are a large number of them in the aerospace industry)

PRODUCTION AND VERIFICATION PHASE

Production and verification is obviously when the product gets made and tested, and ends (hopefully) with a sale to the operational user (customer)



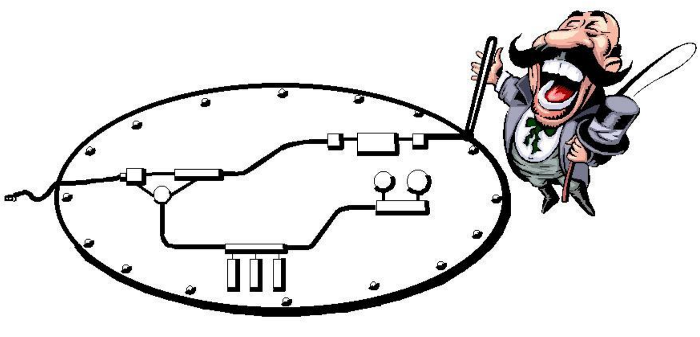
Production activities for the system engineer are similar to the detailed design phase. That is monitoring the impact of the real components as they are constructed on the system. Formal “Change Control” procedures will be in place to examine any proposed changes to manufacturing drawings and system engineers have to analyse the impact of these changes before approval.

System verification activities of the product (that is the actual product to be delivered not the design) should have been planned in principle in the System Design phase. In practice this is often forgotten and they get looked at later – which nearly always the cause of problems. Customers will be looking for tests that prove all the requirements in the requirement specification have been met.

Production and Verification finishes with the delivery of the system. Again this is often a formal review process (with a name like Delivery Review Board) before “handing over the keys” (and collecting the cheque).

UTILISATION PHASE

This is the useful life of the product – when it hopefully does what the requirement specification said it ought to do – and hence a happy customer.



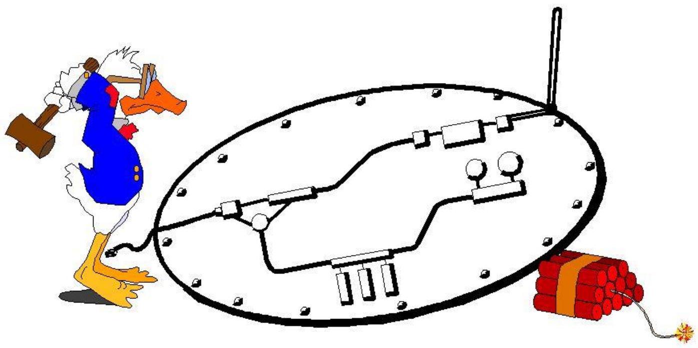
If the system does not meet the specification (normally found in testing during the production phase but subtle problems can sometimes slip through) the system engineering processes will be used to correct the problem. Also the system engineering processes are used in accident investigation and other safety issues.

Much more common is the measuring of the system performance against expectation for general education. Nearly all engineering systems are monitored by the organisation that produced them to explore how they perform in real life. So systems engineers often need to look at the real life performance vs the intended performance. In order to

- Learn from experience
- Examine any shortfalls against expectation; called Validation
  (not the same as failing to meet the requirement specification)
- Examine proposed extensions of requirement (e.g. lifetime)

So customer's utilisation is the start of pre-project phase of the next generation system for the manufacturer; thus completing the product lifecycle.

DECOMMISSION PHASE



The phase everybody seems to forget, getting rid of the system when its useful life is over. Even when it is not forgotten this phase is rarely given the attention it deserves during the development process. For example Decommission is not an official NASA phase (it includes decommissioning in Phase E as can be seen in earlier overhead). However it is very important and requires a different set of study and analysis so we will treat it as a separate phase.

Examples of how to not do it:

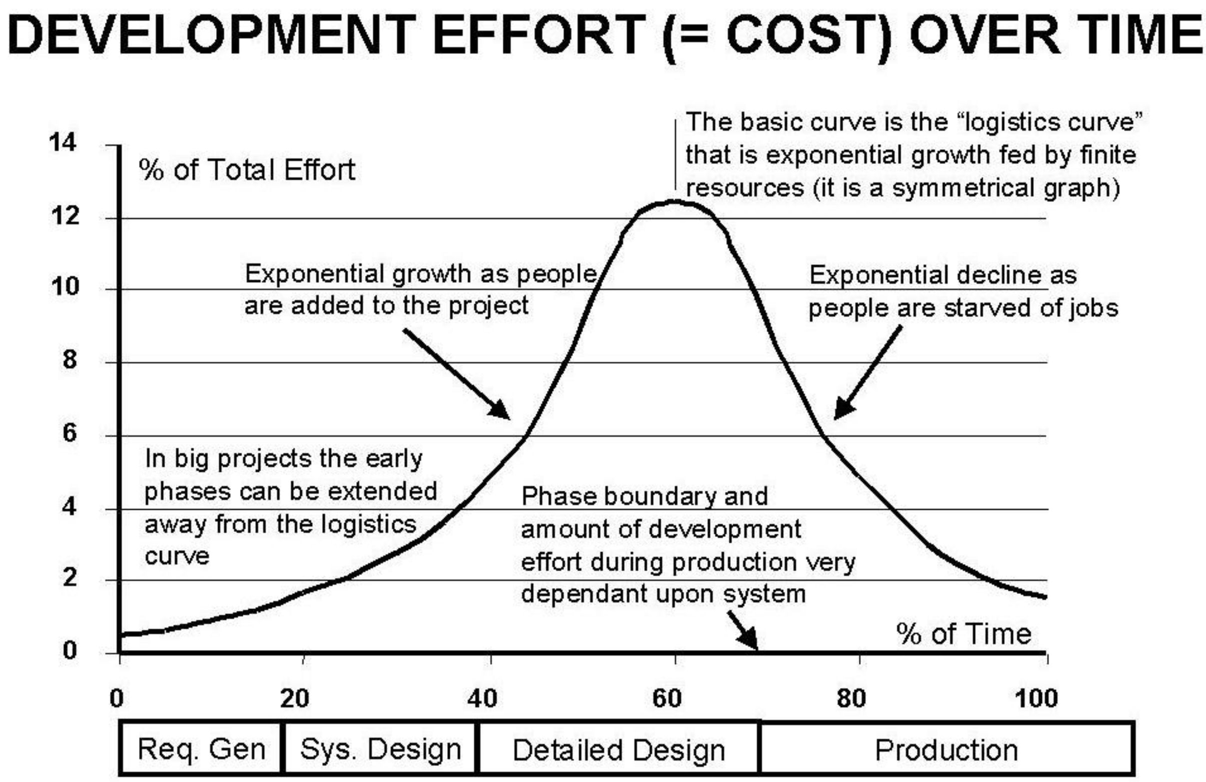
Skylab - Assumed the risk would be accepted if it was allowed to just randomly re-enter and hit the Earth ,which when the time came proved to be unacceptable. They also assumed they would have the Space Shuttle to do something about it which in the end they didn't.

Nuclear power - Ridiculously optimistic assumptions and they did not understand the full implications

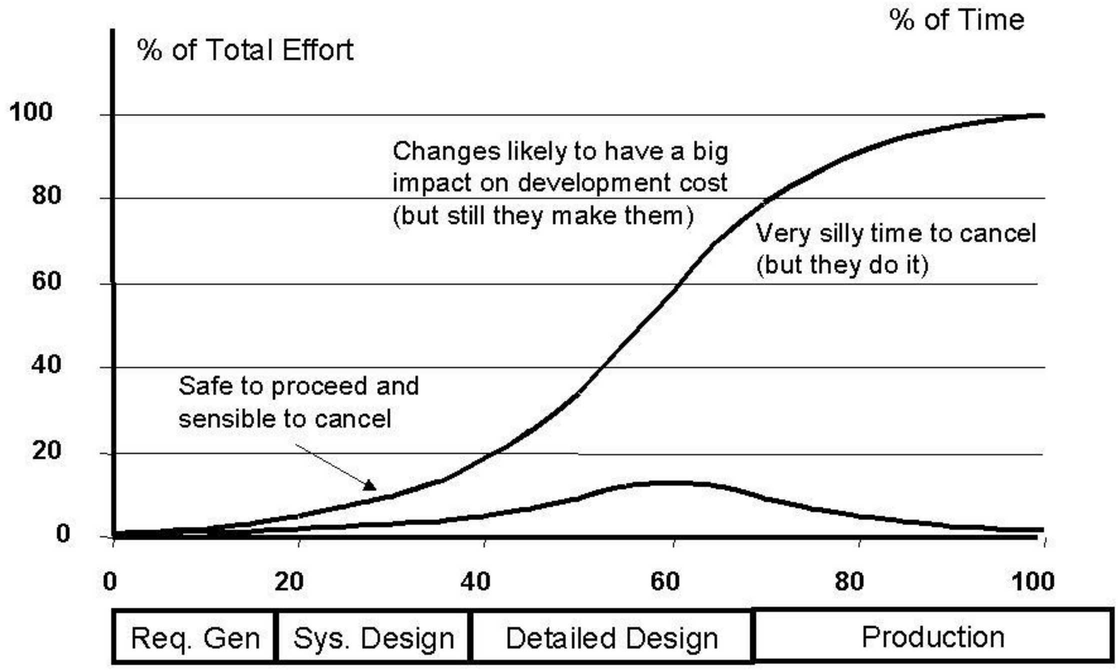
One common issue is an underestimate of the importance of the public perception of risk as well as the real risk.

It is most important that decommissioning is considered during the design phase. To show that the system can be safely disposed of, to explore and resolve any potential problems, AND TO ENSURE THE COST OF DISPOSAL IS INCLUDED IN THE LIFECYCLE COST.

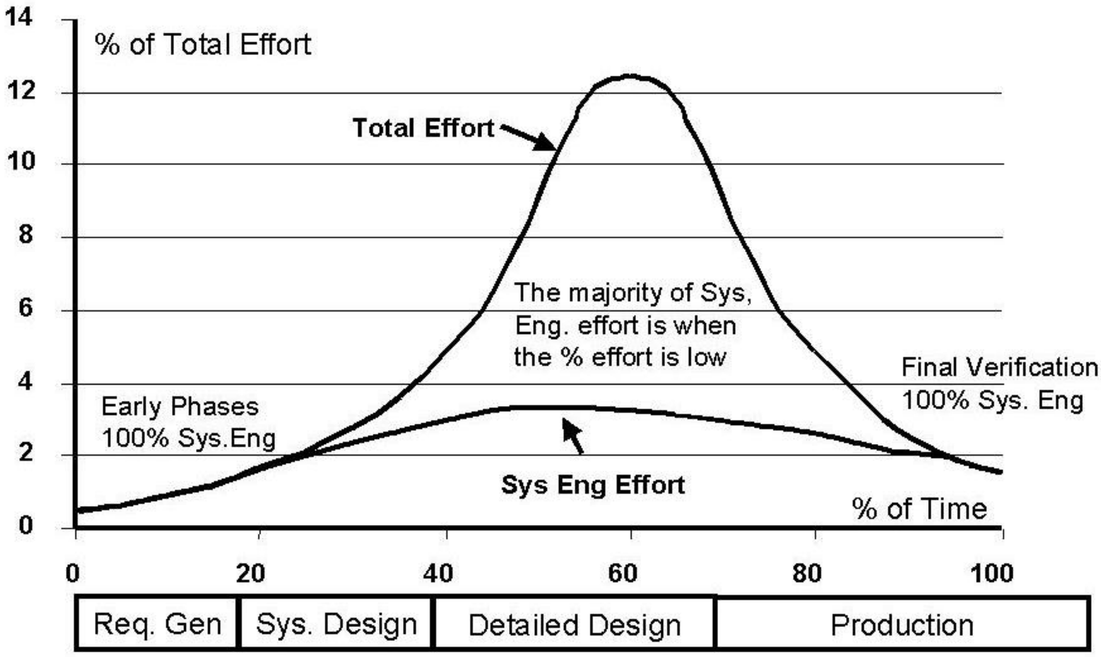
EFFORT (= COST) DURING DEVELOPMENT PHASES



CUMULATIVE EFFORT (= COST) OVER TIME



SYSTEMS ENG. AS PART OF TOTAL EFFORT



REAL LIFE INTRUDES

This lecture outlines the theory of how it should be done and it is normally done a good idea to stick to it. In practice almost no project ends up doing it properly - the reasons:

- Most people involved have not had formal system engineering training and only learn by experience therefore the theory is not understood.

- Budget delays etc. mean optimum programme cannot be implemented.

- Over-optimism on what can be achieved especially in the requirement generation phase.

- Not reading the organisation's procedure that defines the lifecycle phases.

Bad signs are:

- Funny phase names (e.g. in the space industry Phase A2, Phase B prime) and other complications to the classic development sequence.

- Failing to spend 5% to 8% of development cost before freezing the requirement specification.

- Failure to freeze output of development phases.

The last is the biggest sin and also the most frequent.