

# **Project Development and Company Operation Principles and Practices**

Written by :- Mr Kim Lyon  
Managing Director  
World Computers Ltd. ( U.K. )

Web Site :- [wrlldcomp.com](http://wrlldcomp.com) .

Copyright (C) : World Computers Ltd. ( U.K. ) . 2000 - 2008 . All rights reserved !

This material is subject to the proprietary rights of World Computers Ltd. ( U.K. ) . It may only be used with written permission from Mr Kim Lyon . Acknowledgement of source and copyright ownership must be made to this document and to the author and company . All rights remain with World Computers Ltd. .

1.	Basic Business.....	2
1.1	Capital .....	2
1.2	Running Costs.....	3
2.	Company Operation.....	4
2.1	Basic Management Techniques.....	4
2.2	Basic Business Operation .....	5
2.3	Cultural Basis .....	6
2.4	Operational Working Basis.....	6
2.5	Commercial Working Basis .....	7
2.6	Technical Working Basis.....	7
2.7	Working Conditions .....	7
3.	Project Development.....	8
3.1	Basic Techniques .....	9
3.2	Responsibilities of the Project Manager.....	10
3.3	Maximising Capital Productivity - the Capital Value to Capital Cost Ratio .....	12
3.4	Lessons from Life .....	<b>Error! Bookmark not defined.</b>

## 1. Basic Business

### 1.1 Capital

The most important principle in business is to treat the product - the software being developed - and the staff as assets - to apply the business principles concerning assets to both of these items .

The success or failure of a company is closely related to whether the staff understand the basic elements of business - basic Business 101 . It is often amazing in how many companies that these basic principles are not properly understood and acted upon .

The fundamental aspect - requirement - of all companies is ' capital productivity ' . It is off this that everything comes . Capital - money invested - is put into the company where it becomes ' equity ' - ownership and control over ' assets ' . The ' capital value ' of a company should always be the same or , ideally , should increase .

Within a company assets are , typically :-

- 1) physical - tangible - assets such as buildings - eg. offices etc. , equipment - eg. computers , plant , machinery etc. .
- 2) semi tangible - product base - within a software development company this is known as the code base .
- 3) more intangible - human - staff assets .

All assets depreciate . As such - if the capital value is to be retained an additional upward pressure - an additional input \ incoming has to be applied .

Some of the factors to take into consideration with these are :-

- 1) physical - tangible - this requires a large upfront cost . The cost of these assets should be minimised . One classic way of doing this is to ensure that a large percentage of staff work from home .
- 2) semi tangible - the code base . This is where a lot of companies go wrong - in not understanding that it is an asset and dealing with it - in it's development - in how it's developed - as an asset . If a company continuously almost completely redevelops it's code then the asset value of it's code is low - it's cost is high - it's profit potential is low . If , however , the company builds on top of it's code base - designs it with a good architecture and continues to update and expand it then it's asset value is maintained and increased . There must be ' maximum bangs per buck ' . Code must have ' maximum locus of validity ' - ' maximum market reach ' - in terms of both spatial and temporal coverage . The cost of developing , maintaining and updating the code - in the long term - must be minimised . These maximise asset value , minimise depreciation and maximise profits .
- 3) more intangible - staff assets . If the ability , usage and performance of the staff is maximised the value of the staff , as an asset , is maximised - they appreciate in value . If , however , they are commoditised they lose their value . A staff member as an asset is far more cost effective than a staff member as a commodity - applied intelligence , commonsense and wiseness within a good professional , moral and ethical atmosphere is far more cost effective and economically productive than pure grunt .

## 1.2 Running Costs

Within any business there are items that do not make any profits - they are running costs - overheads etc. . It is very important that these are minimised . In a more conventional sense overheads are viewed as items such as electricity , communications , water , secretarial , accounting etc. . However the concept should be extended much further to items such as :-

- 1) inefficiencies designed into the architecture of the code - badly designed code that introduces the requirement for much more labour than should be required in using it .
- 2) inefficient management - work - cultures . Cultures that don't get the most out of the staff - cultures that demotivate the staff , that require the staff to suspend large amounts of their intellect and ability . Badly , usually deliberately badly , organised chains of command . Lack of engendering , permitting and requiring responsibility . Etc. Etc. Etc. . The staff say ' why bother ?' - and , often , don't .
- 3) the pushing of personal and political agendas within the work place . It's surprising how often this occurs and the many forms it can take . This is corrupting of the workplace . Again - the staff say ' why bother ?' - and , often , don't .

The basics are the foundation on which the business is built . If you don't get the basics right everything else will fall down around them .

## **2. Company Operation**

The company is essentially a creative company . As such the company is essentially a very different beast from grunt based companies . The company operates on the basis of clearly separating it's creative side from the grunt side operations of it's clients .

### **2.1 Basic Management Techniques**

The success of a company is heavily effected by how well the tool face - the sharp end - where all the product development is carried out - is managed . To just plug in a management technique and expect everything to be magically sorted is just plain simplistic . It is very important to be aware of what the situations are and to handle them in situ . It is also important to keep things simple but not simplistic .

The basic rule of good management is " Do unto others as you would have them do unto you " . Always put yourself in the other person's position as that other person - where he or she comes from and is currently - and examine the results you predict of your proposed actions . If you find that the results are wrong then your actions are wrong .

It is always good business to act in a moral manner - " What goes around comes around " . Morality is not so much about imposing morality on other people but rather imposing it on yourself .

Don't " Throw the baby out with the bath water " - it is very important to maximise staff's ability . No one is perfect and likewise no one is totally imperfect . Directions must always be positive - it's not that someone is doing something wrong but that they are not doing something right .

Always remember - perceptions are determined by assumptions - if you make the wrong assumptions you will miss things or you will see things that are not there - be careful that you do not delude yourself and make poor judgements .

Always be aware of ' locus's of validity ' - basic set theory - basic theory of knowledge - the validity of a statement is determined by the locus of validity of that statement - the narrower - smaller - the locus the more invalid it is .

Always remember - " just because you don't see it doesn't mean it isn't there " - you need to scratch below the surface and see what really is or isn't there .

Don't assume that everyone else is the same as yourself - they aren't .

Get to know yourself and life . Always - think ! think ! think ! and challenge yourself .

Always be professional - don't let personal issues enter the workplace - these are luxuries you can't afford . Be objective . Stand back from the situations and see them for what they are . Use philosophical and wiseness tools .

Always operate on a self contained basis - this allows problems to be contained , isolated and dealt with . It builds problems out of systems rather than into them .

Always be civilised and mature . Don't be soul destroying - don't attack the soul . Always be responsible - handle responsibility and give out responsibility - delegate .

Trust in people - believe in them - have faith in them , support them and help them to achieve their aims . Support people who go out of their way for you - don't ever give cause for a person to say " Why bother ? " .

Value opinion and diversity - these are strengths in a business . Make sure the overall picture is covered but also take into account changing market situations . Deal in reality .

Always keep good communication flowing in both directions - know what is happening and make sure that the staff know what you want to happen .

It is important to distinguish between naiveté and ignorance . Naiveté is the state of not knowing - ie. you are naive about things you don't know - it is not wrong . Ignorance is the state of refusing to know - it is wrong. Naiveté must be dealt with by education - ie. by spreading around knowledge , by asking questions and getting answers . Ignorance , however , must be avoided .

## 2.2 Basic Business Operation

The basic aspect of company operation is to maximise what is called ' Capital Productivity ' . Capital Productivity is maximising the asset value of the company - what is put into the company , as an investment , and how much the value of this investment is maintained and increased . As such the following factors come into play :-

- 1) product development costs - this is largely a combination of good product design and good development management .
- 2) revenue - profit margin - sale price along with the manufacturing and distribution costs - and - product market size - how large the market is .
- 3) product life - how long the software , in this case , lives . Software typically has a median life of about 5 to 8 years . as such it depreciates in value by 12 to 20 % per year . Software lifetimes of 8 years or more are quite achievable and should be aimed for . The software has to be well designed in order to maximise it's life . This is largely a componentisation issue .
- 4) how much work is put in upfront - if a project is developed in a ' quick and nasty ' fashion this actually builds in a lot of downstream costs . If extra work is put into the project initially these downstream costs can be largely negated . This is both a componentisation issue and a re-configurability issue . Management techniques can also impact strongly on this .
- 5) how much the existing work is built on - it is very important that products be continually updated - that existing market be maintained and expanded . Otherwise the value of the existing capital - the existing product base - will be lost .
- 6) the value of the code that is written . The code itself must be of high value per quantity of code . That means it must be as compact and as effective as possible . This comes down to good code design - good application of design philosophy . Good principles and concepts must be used and good coding standards must be used .

### 2.3 Cultural Basis

The cultural basis of the company is the Professional Working Culture . As such :-

- 1) All staff members are expected to always be professional .
- 2) All internal personal issues are to be left out of the work place .
- 3) All staff members are expected to be emotionally self secure – not to base themselves on externals .
- 4) All staff members are expected to always conduct themselves in a decent , civilised , responsible and moral manner .
- 5) The pushing of personal and political agendas within the workplace is not permitted .
- 6) The making of personal attacks by any staff member on any other person within the company or connected with the business being undertaken by the company – either as a supplier or as a customer - is not permitted . Political , personal and internecine warfare within the workplace are not permitted .
- 7) The loyalty of all the staff is required , at all times , to be to the company as a whole and , as such , to all staff members of the company as a whole . The best interests , particularly in the long time , of the company and of the company's clients must be kept in mind at all times .
- 8) The company is essentially communitarian in nature . This being individuals operating within communities . Communities supporting individuals . Individuals supporting communities . Communities supporting communities . Fully inclusive . Fully respective . As such tribalism must not be present within the company - there must be no ' us and them's ' - and , also , no group think - operating in the company .
- 9) The primary priorities of the staff must , at all times , be :-
  - i) the long term commercial interests of the company .
  - ii) doing the best possible job – in it's entirety .

### 2.4 Operational Working Basis

The operational working basis of the company is :-

- 1) all staff must be operate in a non discriminatory manner – all staff must be taken on purely on the basis of whether they can do the job required of them in it's entirety . The most suitable and best qualified ( academically and experience wise ) staff must be offered the positions . Where it effects the company - the moral integrity of the staff must also be taken into account .
- 2) the moral basis of the company is Christianity particularly within the Western Gnostic and Catholic context . This is stated as a guiding principle .
- 3) corruption of all types – small c ( moral corruption ) and big C ( financial etc. corruption ) , whether given or received , is prohibited .
- 4) gifts may not be given by any staff member to any other staff member . Commercial gifts must also be avoided - the giving or receiving of . Staff must not be corrupted or be put into a position where they are forced to be corrupted .
- 5) there is a general no fraternising rule between staff members . Staff members may go out for a drink or meal together during the working day but staff members may not invite other staff members out to non work related events – dinner etc. . Staff members must not put other staff members into a compromising position .

## **2.5 Commercial Working Basis**

The commercial working basis of the company is :-

- 1) the capital productivity – particularly the capital value to capital cost ratio , over the long term , must be maximised .
- 2) sensible decisions must be made - such as :-
  - i) where it is more cost effective to employ a single more skilled person in place of two or more less skilled people the single more skilled person must be employed .
  - ii) where componentisation and architecture designs can result in costs savings these must be used .

## **2.6 Technical Working Basis**

The technical working basis of the company is :-

- 1) design philosophy and particularly theory of knowledge must be used – the ‘ locus of validity ‘ must be maximised and ‘ superior truths ‘ must be pursued .
- 2) Simplistic but big complicated messes - must be avoided at all times rather – simple but sophisticated must be pursued in design .
- 3) All projects must be preceded by requirement specifications , supported by full design documentation – including full architecture and interface specifications – and finished off with test specifications and test results . Testing must , if at all possible , be undertaken via automated test harnesses with the results logged . There must be a one to one correspondence between the business requirements , the technical requirements , the technical design and the test results .

## **2.7 Working Conditions**

The working conditions of the company are :-

- 1) the standard number of hours are 1748 working hours per year .
- 2) hours can be worked flexibly . Starting and finishing times , providing core and meeting times are adhered to , can be varied . Hours can be worked up and then time subsequently taken off .
- 3) a minimum of 2 weeks per 6 months of leave must be taken off .
- 4) payment to staff must be made on actual value – not on depreciated market conditions . Where appropriate there is to be a profit share made to permanent staff members . This is to be decided , on an individual basis , by the board .

### **3. Project Development**

The project is to be designed using formal methods . Under this the Project Manager must undertake the following steps :-

- 1) Put together a requirement specification for the whole project . A requirement specification is a fairly general specification that is designed to outline how the whole project basically goes together and what it's basic functionality is - what is required of the project .
- 2) Design the project architecture . This shows the relationships between the various modules and components within the project .
- 3) Allocate the modules to the individual engineers . The individual engineers will :-
  - i) Put together a set of specifications for the components - inputs , operations and outputs .
  - ii) Specify the linkage of the components to the module .
  - iii) Specify the Application Program Interfaces . The main ( module ) API must be specified by the Project Manager .
  - iv) Design and develop the individual components .
  - v) Put together a development report for the project .
- 4) Put together a set of test specifications for the modules .
- 5) Arrange for the test engineers , or the individual programmers , to put together a set of test harnesses .
- 6) Ensure that the modules are tested and the results are recorded .
- 7) During the development - handle any issues brought up by the engineers .

A very good method of running project development is to use Headline Points - Event Points . This allows the project to be broken down into the basic individual requirements on a point by point basis .

Architecture , componentisation and coding standards ( for software development ) are all very important . They impact on general project design issues and on code maintainability and upgrade ability . It is very important that the project is designed in such a manner that the maintenance ( updates , reconfiguration and bug fixes ) costs are minimised and that the life of the software is maximised . The project should be designed in such a manner that whenever changes are carried out only minimal changes are necessary and the changes propagate through the project thus updating anything associated . This means that relationships must be set up - such as using enumeration's , macros and typedef's . Also table drive as much as possible . IE. don't forget the Theory of Knowledge - maximise the Locus of Validity . Don't hard code . Use single definitions . Use as much commonality as possible . Keep things simple but not simplistic . Simple is optimal . Simplistic can , in fact , end up making things much more complex .

Communication is very important . It is important to maintain full and effective communication with all the staff . Have regular - daily - brief meetings - 10 minutes max. - and get a short - 2 minutes max. - report from each staff member on what was achieved the previous day and on what the aims are for the current day . This helps keep everything on course . Don't forget - it's your responsibility to ensure that the project directions and the corresponding aims are clearly understood and achieved . The staff have to know where you want to go and you have to make sure that they are achieving those aims . If they are not you have to find out why and then provide them with the facilities such that they can achieve the aims . It is important to be constructive and to assume that the staff want to achieve the aims .

Documentation is very important . The documentation must be clear and complete . Do not write in context . Spell everything out clearly . It is vitally important that the documents can be read easily and quickly . Documents should have the document name and number ( within the document data base ) at the top left hand position . The page number should be placed at the top right hand position . Page no. of no. of pages should be placed at the bottom middle . If the document has a single author the author's name should be placed at the bottom left . Alternatively the document number can be placed at the bottom left .

It is very important to maintain a bi-directional documentation chain - setting of specifications and documenting that the specifications have been met and how they have been met - drill down and report back .

The architecture should be developed in consultation with the development engineers . It should very much be the Project Manager or Architect's job to specify the architecture , the interfaces , the overall operation and the functionality of the components .

The Architect should be looking at the components from an external point of view . The development engineers should be looking at the components from an internal point of view .

Code wise - the code should be heavily commented . It is a good idea to put details of the project design with the code as the code can often become separated from any associated documents .

The project must be organised logically - in a tree format and with common components grouped . The project documentation must be associated with the project and details of the project architecture must be placed within the main project files .

### 3.1 Basic Techniques

The basic techniques - modus operandi - should be :-

- 1) in developing the design the design should be drilled down . The major components - branches should be constructed - top down . The branches should then be fleshed out - bottom out . the drilling down - the comments - should be used as the basis of the construction .
- 2) componentisation should be used - the project should be divided into component areas and each component should be entirely responsible for it's individual area - data and functionality being encapsulated . If at all possible an Object Orientated language - such as C++ - should be used .
- 3) at strategic points components should be dynamically bound and should be operable on the basis that if they are included in the project operation - started up - they are then automatically part of the project operation and , vice versa , if they are not started up - they are then automatically not part of the project operation .  
each component :-
  - i) when started up will register themselves with the component they are dynamically binding to - pass a pointer to their dynamic binding table and , if required , to the components characteristic table .
  - or
  - ii) on the start up function being called being supplied with a pointer to the binding points dynamic binding table and returning a pointer to the component's dynamic binding table and , if required , to the components characteristic table .
- 4) asynchronous techniques should be used - event queues \ mailboxes . These design out the problems associated with synchronous techniques - mutexes , semaphores , event flags etc. - producing lock ups etc. .
- 5) the design should be integrated as much as possible - eg. the dynamic binding linkages - addresses - should be used as part of the event message addresses .
- 6) the design should be on the basis of rules - commonalities - and the use of basic principles and concepts - not on the basis of exceptions .
- 7) the design should be heavily instrumented with asserts . The asserts should check for expected conditions and assert on these conditions being false .
- 8) the components individually and the project as a whole should be tested by using automated test harnesses . Non test harness based testing - human based testing - should only be used where a test harness can't be used - eg. overall in situ UI testing .

### 3.2 Responsibilities of the Project Manager

The responsibilities of the Project Manager are :-

- 1) the buck stops with the Project Manager . The Project Manager is responsible for ensuring the successful completion of the project design .
- 2) to ensure that the long term commercial interests of the company are protected .
- 3) to , on the basis of a clear project description , develop a set of requirement specifications for the project . The project must be properly planned . The requirement specifications should consist of a combination of bullet points and component requirements .
- 4) to ensure that all the various stages of the project are scheduled correctly .
- 5) to resource all the suitable people required for the development . The Project Manager must have a good and valid knowledge of people and life . The Project Manager must be socially skilled - in the broadest sense . The staff must be individually self contained - internally based - and must be morally strong - and able to work with a wide variety of people . The company must have a good moral and individual base . Strength is in diversity .
- 6) to ensure that the staff are able to meet all the requirements of the project - on a professional and individual basis . They must have the ruggedness of training and ability to be able to tackle the project fully and to be able to do a proper job .
- 7) to ensure that all issues effecting the project development are properly prioritised . High cost luxuries ( such as political and personal agendas ) must be avoided - very often these luxuries sink a project and a company . The most important thing is the project design and the consequent profitability of the company .
- 8) to provide all the necessary equipment , tools , books etc. to do the job . These must be supplied as and where needed .
- 9) to always keep in touch with the reality of the situation . To always be truthful . Not to delude . To say things as they are . To provide constructive and helpful feedback .
- 10) to make sensible expenditure decisions . EG. don't try to save on equipment cost if this results in far greater labour cost . Expenditure must be appropriate , planned and managed . Likewise with staff costs - a more capable engineer will usually be very cost effective - over multiple less capable engineers . Projects do not scale up cost wise -  $1 + 1 \neq 2$  performance wise due to the management overhead .
- 11) to provide full support to the staff - to ensure that they can do their job . People aren't perfect - where people require extra knowledge or skills , in order to complete the job - these must be provided by the Project Manager to the staff member - usually indirectly . Staff must not be personally attacked . The loyalty of all the staff must be to the company - and to all of the staff - as a whole .
- 12) to maximise the validity of the approach . To ensure that the best possible design methods are used . Fire fighting should not be used . Problems should be dealt with . This is primarily a design philosophy , morality and wise ness issue .
- 13) to ensure that the project is designed in a componentised manner - where the re-usability of the components is maximised - and with an architecture - where the versatility of the system is maximised . Bugs should be contained , identified and eliminated - not hidden in interactions .
- 14) to ensure that the project is fully and proportionally documented . Specifications and project documentation - of all types and at all levels must be maintained . All issues raised and their associated addressing must be logged . A full chain of accountability - in both directions in a documented form - must be maintained .
- 15) to ensure that all communication is effective . People can't read minds . Working assumptions and basis' must be stated . Requirements and specifications must be clearly stated . If something hasn't been stated it mustn't be assumed by the Project Manager to be there . The Project Manager must ensure that everything is clearly understood .
- 16) to hold regular ( daily ) meetings ( no more than 10 minutes long ) and to make frequent checks on the progress of the project and to indicate where changes need to be made . This is purely a technical exercise and must be handled as such .
- 17) to test the project - or to ensure that it is sufficiently tested - and to ensure that it performs satisfactorily . To ensure that the various components and modules function satisfactorily of themselves .
- 18) to ensure that the various components and modules integrate satisfactorily .
- 19) to manage the ownership of responsibility and consequent interactions of the various parts of the project staff wise .
- 20) to ensure that the project can be easily maintained , updated , evolved and expanded .

- 21) to ensure that the project development is not effected by politics or personalities . To protect the staff from the side effects of politics and personalities . The work place must be a professional - 'ism free - zone . The pushing of political and personal agendas must be prohibited . Staff have to be able to concentrate on the job - they mustn't have their time wasted and they mustn't be de-motivated . It is very important to maximise the contribution of the staff .
- 22) to ensure that corrupting influences are kept out of the project development . There must be a ' don't screw the crew ' rule . Personal items such as the giving of cards , presents , donations and dinner invitations must not be allowed . Ego stroking is strictly prohibited . People must not be required to be ' emotional prostitutes ' . There must be no social engineering . There must be no " Us and Them " - no tribalism \ elitism and no gutter crawling .
- 23) to ensure that there is no Feudalism .
- 24) to ensure that there is no Tribalism . There must be no ' us and them's ' operating in the company . The company is communitarian . This being individuals operating within communities . Communities supporting individuals . Individuals supporting communities . Communities supporting communities .
- 25) to ensure that the company operates on a community basis – that everyone supports everyone else . Politics and personalities are not permitted . There must be no emoting . Decisions must purely be made on technical and business grounds – no other . Loyalty must be to the company as a whole .
- 26) to ensure that the Professional working culture is used . The Professional culture is tried and proven – it has been used for very many years very successfully and is the culture that must be used . All staff are required to be professionals and to operate in a professional manner . Staff are expected not to allow internal personal issues to impinge on their work situation .
- 27) to ensure that the staff are given areas of responsibility – ownership .
- 28) the Project Managers must develop the overall framework – structure – of the project – then take on the necessary staff to flesh out the project . This is an architectural design and development and a technical management exercise . If necessary the Project Manager may employ an architect .
- 29) to ensure that all the staff work together - in a collaborative manner . Each staff member must be made aware of what is technically required . The project manager must ensure that the project components are designed and that they all fit together .
- 30) to ensure that diversity is maintained . To ensure that dissidence is allowed and supported . Dissidents are often the most loyal of staff members - they see the reality and try to ensure that things are done right .
- 31) to ensure that all staff are able to operate as individuals . They must be able to make any statement regarding the work being done without fear of censure - ie. within the professional domain - a complete free speech environment is to be encouraged and maintained .
- 32) to ensure that there is no discrimination - discrimination , which , in it's nature , attacks aspects of diversity , attacks the maximising of the commercial competitiveness of the company . Any discrimination - positive or negative is positive and negative and as is prohibited . All staff must be supported .
- 33) to ensure that the staff are treated as professional human beings and are not be used and abused . There must be a distinct separation between the staff member's professional life and personal life .
- 34) to ensure that the staff don't under or over work . The staff must , at all times , be at their peak performance level . They must be able to stand back from the project and to have an overall view of - perspective on - the project . This requires that the staff must be able to have the time and energy ( along with the ability ) to be contemplative - to have perspective of the project and to be able to think the project through .
- 35) to ensure that staff are able to work on a flexible basis . Within certain core hour requirements staff must be able to set their own hours . Where appropriate remote working must be accommodated . Where staff are remote working and where they have not worked on a project before for the company they may be required to work 1 or 2 weeks in every 4 to 6 weeks in the office - as is necessary . This may also be required for subsequent projects .
- 36) to ensure that the project design is carried out in a secure manner .
- 37) The Project Manager must always act in a professional , moral , fair , just , loyal and non corrupt manner . Staff must never have to ask themselves the question " Why bother ? " .

### 3.3 Maximising Capital Productivity - the Capital Value to Capital Cost Ratio

When capital is invested in a company it becomes equity - the holding of assets . Some of this capital is used for running costs - overheads - these always need to be minimised as they have no capital - asset - value . The rest is invested in assets . Software that is developed within the company should be treated as an asset - ie. maximised in value and minimised in depreciation . It should not be treated as a running cost - if it is then the company is being run badly and will suffer as such .

It is vitally important to maximise the Capital Value to Capital Cost ratio of the software . As such :-

- 1) All development work must , over the long term minimise the cost of the software - the amount that the software costs to develop , maintain and update . Investment should be made upfront in order to reduce downstream costs .
- 2) All development work must , over the long term maximise the value of the software - the amount that the software earns .
- 3) The cost of the development of the individual components must be recorded .
- 4) The amount that the individual components earn must be recorded .
- 5) The resulting Capital Value to Capital Cost ratio of the components must be calculated , charted and must be regularly reviewed .
- 6) The resulting depreciation of the value of the components must be calculated , charted and must be regularly reviewed .

By doing this the commercial success of the company can be maximised .

I think the best example of how to not maximise capital productivity - how things are disastrously done wrong - are what I call ' idiot companies ' . These usually have the following characteristics :-

- 1) They produce what is essentially regarded as ' customised applications ' .
- 2) Their management have no idea of the basic elements of business - or if they do , supposedly , they have no idea of how to translate these into engineering practice . Particularly they do not understand capital - they are ignorant - blocked off - of the full aspects of capital .
- 3) They sell - license - their product on a per customer - fully customised - basis - where the code is massively redeveloped for each new customer - ie. the code is not designed to be easily reconfigurable - they don't work to a common code base . As a result they have no capital value in the code - their capital value is purely in expertise . Further - the massive redevelopment of the code with each new customer is a massive cost factor - this critically effects the company operation .
- 4) They are usually tribal - ' us and them ' . They are often ' they think they are hot shit but are just a bunch of cold farts warming up ' . They usually attack anyone who makes sensible and essential suggestions - ie. someone who has actually shown more loyalty than they have . They assume that people are enemies when in fact they are not . They can often be quite nasty and vicious .
- 5) They often have large revenues - 100's of millions - but very little profit and even often losses . This is largely because of their lack of understanding of capital . But it can also be additionally caused by them pissing off their staff so much that their staff say ' why bother ? ' and can't answer the question in the positive .

Capital is translated into assets . An asset is something the a company holds that generates income . Assets include the software base . Assets depreciate - lose their value . The software base depreciates - has a limited life . The life of software is critical .

Short life software bases - eg. with a life of 1 customer only - have very little asset value . As a result costs are high and profits are low . When the unexpected strikes , as it does , the profits are often negative . As a result , these ' idiot companies ' are very poor performers - they are hampered by very bad management - usually this is by the management letting personal ( internal ) issues greatly effect their work .