Keith On . . .

Engineering Design

K.E. Schubert

Founder
Renaissance Research Labs

Associate Professor
Department of Electrical and Computer Engineering
School of Engineering and Computer Science
Baylor University

Contents

Chapter 1

What is Engineering?

1.1 Fun Quotes

Thanks to Harry T. Roman of East Orange, N.J., USA, who compiled the following 21 definitions of Engineering.

> The application of science to the common purpose of life.
>
> Count Rumford (1799)

> Engineering is the art of directing the great sources of power in nature for the use and convenience of man.
>
> Thomas Tredgold (1828)

> It would be well if engineering were less generally thought of, and even defined, as the art of constructing. In a certain sense it is rather the art of not constructing; or, to define it rudely but not inaptly, it is the art of doing that well with one dollar which any bungler can be with two after a fashion.
>
> A. M. Wellington (1887)

> Engineering is the art of organizing and directing men and controlling the forces and materials of nature for the benefit of the human race.
>
> Henry G. Stott (1907)

> Engineering is the science of economy, of conserving the energy, kinetic and potential, provided and stored up by nature for the use of man. It is the business of engineering to utilize this energy to the best advantage, so that there may be the least possible waste.

Willard A. Smith (1908)

Engineering is the conscious application of science to the problems of economic production.

H. P. Gillette (1910)

Engineering is the art or science of utilizing, directing or instructing others in the utilization of the principles, forces, properties and substance of nature in the production, manufacture, construction, operation and use of things ... or of means, methods, machines, devices and structures ...

Alfred W. Kiddle (1920)

Engineering is the practice of safe and economic application of the scientific laws governing the forces and materials of nature by means of organization, design and construction, for the general benefit of mankind.

S. E. Lindsay (1920)

Engineering is an activity other than purely manual and physical work which brings about the utilization of the materials and laws of nature for the good of humanity.

R. E. Hellmund (1929)

Engineering is the science and art of efficient dealing with materials and forces ... it involves the most economic design and execution ... assuring, when properly performed, the most advantageous combination of accuracy, safety, durability, speed, simplicity, efficiency, and economy possible for the conditions of design and service.

J. A. L. Waddell, Frank W. Skinner, and H. E. Wessman (1933)

Engineering is the professional and systematic application of science to the efficient utilization of natural resources to produce wealth.

T. J. Hoover and J. C. L. Fish (1941)

The activity characteristic of professional engineering is the design of structures, machines, circuits, or processes, or of combinations of these elements into systems or plants and the analysis and prediction of their performance and costs under specified working conditions.

M. P. O'Brien (1954)

The ideal engineer is a composite ... He is not a scientist, he is not a mathematician, he is not a sociologist or a writer; but he may use the knowledge and techniques of any or all of these disciplines in solving engineering problems.

N. W. Dougherty (1955)

Engineers participate in the activities which make the resources of nature available in a form beneficial to man and provide systems which will perform optimally and economically.

L. M. K. Boelter (1957)

The engineer is the key figure in the material progress of the world. It is his engineering that makes a reality of the potential value of science by translating scientific knowledge into tools, resources, energy and labor to bring them into the service of man ... To make contributions of this kind the engineer requires the imagination to visualize the needs of society and to appreciate what is possible as well as the technological and broad social age understanding to bring his vision to reality.

Sir Eric Ashby (1958)

The engineer has been, and is, a maker of history.

James Kip Finch (1960)

Engineering is the profession in which a knowledge of the mathematical and natural sciences gained by study, experience, and practice is applied with judgment to develop ways to utilize, economically, the materials and forces of nature for the benefit of mankind.

Engineers Council for Professional Development (1961/1979)

Engineering is the professional art of applying science to the optimum conversion of natural resources to the benefit of man.

Ralph J. Smith (1962)

> Engineering is not merely knowing and being knowledgeable, like a walking encyclopedia; engineering is not merely analysis; engineering is not merely the possession of the capacity to get elegant solutions to non-existent engineering problems; engineering is practicing the art of the organized forcing of technological change ... Engineers operate at the interface between science and society ...
>
> Dean Gordon Brown; Massachusetts Institute of Technology (1962)

> The story of civilization is, in a sense, the story of engineering - that long and arduous struggle to make the forces of nature work for man's good.
>
> L. Sprague DeCamp (1963)

> Engineering is the art or science of making practical.
>
> Samuel C. Florman (1976)

1.1.1 Class Quotes

> Engineers use vision and resources to create a design that is cost-efficient and maximizes performance.
>
> Jonathan Carranza, Jason Fredrick, Charles Korman, Sarah Wade (2008)

> Engineering is a creative and altruistic way to benefit society using broad knowledge and tools efficiently to design technology.
>
> Maged Assad, Diane Bernal, Erin Powers, and Amber Thomas (2008)

> If you are going to reinvent the wheel, do it as optimally, efficiently as possible... and damnit make a better wheel! And remember when designing embrace the laziness.
>
> Tim Castelli, Steven Parker, and Sachithra Udunuwarage (2008)

1.2 Fields

The "Big Three":

Civil The name comes from civilian rather than military engineering, but is used of large construction projects such as buildings and roads.

Electrical Design of electrical or electronic components. Includes such areas as analog and digital circuits, electro-magnetics, signals, power, controls, estimation. This has many sub-fields such as ours - computer engineering.

Mechanical Design of machines and devices. Typically includes such areas as statics, dynamics, materials, fluids, and thermodynamics. This has many sub-fields such as industrial and aerospace.

1.3 Profession

Engineering is a true profession, as opposed to many areas of study often so called because engineering has a licensing process. To become licensed, you must pass the EIT/FE[1] exam, then work a number of years under a professional engineer (PE) or in certain fields (such a aerospace) or get education credit (ABET accredited schools only)[2], then pass a second exam (PE exam)[3]. Typically continuing education courses are required for renewal.

As a profession, engineering is overseen by state laws and professional societies. Ethics in engineering is thus not an arbitrary concept, it is a professional standard and requirement on our actions.

[1] Engineer-In-Training/Fundamentals of Engineering exam is a one day, 8 hour exam with around a 20% pass rate when I took it.

[2] The number of years required varies by the area you are trying to get a license in.

[3] A 2 day exam of about 8 hours a day.

Chapter 2

Communication

2.1 Weekly Status Report

2.1.1 Team Leaders

Weekly status reports have four basic sections: scheduled tasks, accomplished tasks, meetings, project status, goals/assignments for next period. Much of this revolves around the schedule/Gantt chart. The entire report should be 1-2 pages in length. Shorter is better if it covers the points.

In the first section, you should give a brief statement of what was the goals for the week, and how they align with or derive from the schedule/Gantt chart. The person or people assigned should be included. Brief means enough detail is given to understand what was to happen and how it fits in the larger program. Do not give so much that it takes to much to read (a paragraph or two is normal), or so little that the reader can't judge what was done and how it will impact things. Note that the schedule can get modifications and tasks can get added, deleted, or amended, so it is important to cite the date or revision of the schedule/Gantt chart.

In the next section you should cover what tasks were actually accomplished, and if this met the goals. Be sure to cite key players, as this is how people get recognition and thus raises and promotions. It is the goal of a manager to help your employees improve and get rewards. These should also be brief (read: get to the point), as whoever is reading this will probably be reading several of these. People need to know precisely what has been accomplished, not a narrative on the accomplishment. You can give a short statement on why something was difficult if it is only occasionally done.

The meetings section should state the attendance, reason (main agenda items) and results or conclusions.

Next, the weekly status report should show the progress incrementally and cumulatively, i.e. what progress did you make this week as a percent/fraction of the weeks goals, and what is your total progress as a percent/fraction of the total project to date. This should sum up in a numerical way what has been accomplished this week and where the entire project stands. Including an updated Gantt chart is optional, but can be very illuminating.

Finally, the status report should have the assignments or goals for each member of the team by name, giving the task, numerical estimate of completion, and task title.

2.1.2 Team Members

Each team member needs to send their status reports so that the team leader can send a team status report to their superiors, such as project engineers in industry, or professors and company liaisons in college design programs. Team members have three things they have to report

1. Goals for the week. These are usually assigned by the team leader, or at least with the team leader's cognizance, but you are trying to make their life easy and show them that you understood your tasks.

2. Accomplishments for the week. This could be a percentage of work done for a task that is larger or the whole task if you finished it. Let them know if it is ahead or behind schedule and any positive notes. Be brief (not rambling) but don't cut out important facts. This is a bit of an art, but basically you are trying to let them know the great things you did, and what is going on, but not take so long they don't want to read it. As a good guide try two sentences per task, the first stating what amount is done, the second stating descriptively what was accomplished.

3. Assignments for the next week.

I have included a sample of a good status report in figure 2.1. Note the problem/alteration/roadblock was put in italics to emphasize it.

2.2 Log Book

Log books are a necessary legal documentation of work that are crucial for a variety of cases including patents. To be useful in court the pages must not be removed or inserted (thus no binders). Blank pages cannot be left, and all pages must be used in sequence. You cannot leave any large blank space, as this would allow for future tampering. Items cannot be "whited out" or scribbled over. Any corrections must be done by a single, thin line through the error.

Each entry should have a descriptive heading with the date, and should be legible and understandable. The log book must contain all work done, including: meeting notes, sketches, ideas, concepts, calculations, phone call notes, research notes, test data, etc. Logbooks are primarily a daily record of what you thought and did, what your goals were, and how you were progressing. It ought to be more than a diary of general activity. Make sure you label sketches/drawings. Include relevant dimensions or scale. Include assembly diagram for context. Note block diagrams are particularly useful.

Think of future users, and evaluate your logbook from this perspective. Make sure you add enough detail to the notes of design specifications and estimates to make them

Figure 2.1: Sample Team Member Status Report.

Keith Evan Schubert
Associate Engineer
Widget Project

Goals:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Finish task 3, model system dynamics widget driven by oscillator circuit.

2. Do 20% of task 4, analyze stability of widget.

3. Do task 8, select widget antenna.

Accomplishments:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Task 3, model system dynamics widget driven by oscillator circuit is 100% complete. The modeling took one day longer than anticipated because nonlinearities in the oscillator proved to be too large to ignore, necessitating a more complicated model. The non-linearity enters only as an input, but can be handled by augmenting the states. The results are in technical report TR-00743-23: Widget Dynamics.

2. Task 4, analyze stability of widget is 10% done, as the non-linear states, which handle the inputs, have been analyzed. The non-linear states form a stable, passive manifold that is bounded-input, bounded-output stable. This will still create complexity in calculating the closed loop control law, which might require an extra week to examine robust control methods if the non-linear control proves infeasible.

3. Task 8, select widget antenna is 100% complete. The Antenna Industries, 9" whip antenna model # 91-471128-678 rev B was the best price-performance antenna and is in the middle of its product life.

Assignments:
All tasks are from Widget Project Gantt chart, 4-17-08.

1. Continue task 4, analyze stability of widget. Achievement of 70% completion was the original goal, but *given the alterations in the problem complexity an adjustment to 50% completion is suggested*, which would entail completion up to nonlinear analysis, but not a generation of a candidate control law.

reconstructible later. If you use an outside formula or result, cite the source so others can find it. Make sure you include enough detail to facilitate future users reconstruction of your efforts. Note even your dead ends and why you abandoned them, as this information is of particular assistance. Relevant loose materials can be attached to a logbook if small enough, or they can be cited in a clear distinguishable way, and stored in a different location (such as a notebook or filing cabinet). Identify contacts and communications by name, company, location/number, and date for future reference.

2.3 Presentations

A good talk starts with lots of preparation. You want it to look like it is no work, so you have to do lots of work beforehand so it seems effortless.

You have to pick a theme and one key point for every 10-20 minutes. Everything else must support this. Keep the goal and theme in mind. Make slides that support you, not that steal (or ruin) the show. Find ways to illustrate your talk in several different engagement styles: pictures, stories, thought experiments (gedanken experiments), etc.

Make sure your slides and so on are readable by everyone in the room.

Keep your slides simple- no fancy transitions and effects- as they will not carry over from one machine to another. Honestly talks are either glitzy or substantive (or neither) - be substantive. Substantive talks can be cleanly elegant, but I have not seen one that is glitzy.

Don't put lots of text on the page. You will tend to read it, and so will your audience. You will be boring and they won't be paying attention to anything you say. Use text sparingly in phrases to highlight ideas and keep people on track.

Avoid notes, try to know your speech. Memorization is a plausible alternative, but knowing it means you have moved beyond mere memorization to understanding and an ability to paraphrase it. This will heighten respect and confidence from your audience.

Keep track of the time. No one wants to be bored by a long talk. Stay on track, but look comfortable on doing this. One technique I use is having optional stories or info for slides so I can add or subtract and my audience is none the wiser. I can then use these thoughts in conversations after the presentation is over, if I didn't use them.

The most important slide is the first one, because it has your name. Don't skip over it or move on too fast. I suggest you give a short bio on yourself (1 min or so). One of your main goals is to let people know who you are - it introduces the rest of your work and provides you future opportunities. Don't sound like an egoist, make it a casual introduction so they get to know you. It is a nice icebreaker, and separates experienced speakers from novices.

The next most important section is the intro, followed by the conclusion. Again, most people skip this. Don't. It is your chance to frame how they look at the work. Why do they care? How should the think about this? Introduce your key points- they will need to hear them at least three times to remember them. Make both clear. The intro allows you to set them thinking in whatever way you want. The conclusion is your last chance to clean up their understanding and thoughts, don't miss it.

The first thing people will see is how you are dressed and groomed. To ensure a good talk make sure you dress appropriately and are well groomed. It sounds shallow; I know. This is the first thing people see and it will affect how they think of you[1]. Notice that I said you should dress appropriately, not dress as fancy as you can. You want to look professional, and basically on par to slightly better than your audience. The earlier you are in your career, the better you have to dress to get respect. You should never look shabby. The one exception to the on par rule is if you ever get to the point you are the undisputed expert in a technical audience, or the owner of the company, etc., in which case you can dress nice but causal and it comes off showing your special status. Don't try this if you aren't in the position to pull it off, but if you can, it is a sign of prestige.

Be calm. No really, be calm. This is the single biggest failure: looking nervous, talking fast, skipping things, etc. Know your stuff, then speak clearly, concisely, and authoritatively. Speak at a slightly slower than normal pace, and put pauses for emphasis. Speed and pauses are like font size and white space: a paper that has tiny fonts and no blank space is ugly and unreadable. Similarly a rushed talk with no pauses is displeasing to the ears and will not get its point across.

Don't name drop without an interesting story that relates to the subject at hand, but then do so.

Use consistent terms unless you have a really good reason, like trying to show that terminology is arbitrary, and even then do it sparingly. Inconsistent terminology confuses people. There is almost always an extreme in all directions, so here is the flip: don't make things needlessly consistent. A little variety opens things up to creative ways to examine things and prevents disciplinary myopia (and sometimes even near magical adherence to arbitrary conventions).

If you are scared then pick a few friendly faces out in the audience and move between watching them.

If you are doing a group presentation you should prepare more and practice so the talk doesn't look stiff. Give nice transitions - introduce the next speaker and their area. Shake hands, give a pat on the arm, share a warm smile, or something else that shows you are all friends and colleagues. It will build confidence in your work and make others interested in knowing you. People like friendly people.

2.4 Reports

Engineering reports and technical writing in general are unique beasts. Many jokes exist about how bad engineers write, some of which has to do with our emphasis on technical correctness over style, but some has to do with what engineering reports are. We are not trying to be pretty, flashy, or entertaining in a report, we are trying to say accurate statements both concisely and atomically[2]. Concisely means to say something correctly in the most efficient way (read least number of words). We value a short, information-packed

[1] I am not arguing if this is right or wrong, but rather I am saying you are foolish not to take such a simple step to improve how people view you.

[2] I am stealing the term atomic from computer science and adapting it to writing.

statement that is still understandable. By atomically, I mean each paragraph must be able to be read independent of the rest of the report, because that is what will happen. People will just look up the sub-section that has a title that makes them think it has the information they want, read that or parts of it till they find the information to use. You need to write with this in mind. Much of this is contained in the following quote (it only misses the random access part).

> How to make engineers write concisely with sentences? By combining journalism with the technical report format. In a newspaper article, the paragraphs are ordered by importance, so that the reader can stop reading the article at whatever point they lose interest, knowing that the part they have read was more important than the part left unread.
>
> State your message in one sentence. That is your title. Write one paragraph justifying the message. That is your abstract. Circle each phrase in the abstract that needs clarification or more context. Write a paragraph or two for each such phrase. That is the body of your report. Identify each sentence in the body that needs clarification and write a paragraph or two in the appendix. Include your contact information for readers who require further detail.
>
> – William A. Wood (email), September 8, 2005

Chapter 3

Systems Engineering and Project Management

Systems engineering is the field of engineering that is concerned with the design and management of engineering products over their lifecycle. Systems engineers thus handle the high level design and oversee the process of design, manufacture, and maintainence. Systems engineering has numerous standards and models to describe the processes used, such as [2, 3, 8, 9, 11].

3.1 Classical Techniques

Probably the most famous techniques are the Waterfall method and the 'V' method. Both are very similar. The Waterfall technique is based on the idea that one phase flows to the next. It develops a systems life cycle that we will discuss below. The 'V' method is similar, but folds the testing in the second half of the life cycle up to pair it with the spec development of the first half. Thus the detailed designs at the end pair with unit tests of the parts that were designed, and the customer requirements and the very beginning pair with acceptance tests. Classical techniques are only done on major military projectes that require such a high level of oversight and no easy breakdown is possible.

3.1.1 System Life Cycle

Any product starts as a need or idea in someone's mind. This is where the initial formulation is done. This is communicated orally and discussed informally until the originator(s) feel it is sufficiently important to formally propose it. The formal proposal is usually called a "white paper". There is no formal specification of a white paper, but it should have an executive summary (1 page overview) at the start, followed by background information, need justification, and a basic concept. If approved the design will go into a formal life cycle. Having worked for a military contractor (Northrop-Grumman), I will follow the military systems engineering life cycle, see the Defense Acquisition Guidebook [4] or Defense System Software Development [6].

3.1.2 Concept Refinement Phase

This is the conceptual design. A Preliminary System Specification is generated (required capabilities, and a systems engineering plan put in place. System safety analysis is begun and test and evaluation strategies are made. Support and maintenance issues are outlined. Alternatives to be analyzed are thought up, a technology development strategy is drafted, and a draft of the capability development document is created. Initial costs and manpower estimates are made.

At the conclusion of the phase there is the Initial Technology Review (ITR), and the Alternative System Review (ASR). The ITR is a review of the technical baseline to ensure there is sufficient development to make a valid initial cost estimate. The ASR reviews the alternatives considered, and assesses if the set of requirements agrees with the customer's needs. For our purposes, most of the local companies have done this before assigning the tasks to our teams. Completion of this phase is milestone A.

3.1.3 Technology Development Phase

This is the configuration design. Technology selection and risk reduction are primary goals of this phase. Further work on costing and systems engineering, resulting in the Systems Engineering Plan, and the Test and Evaluation Master Plan. A Preliminary System Performance Specification is drafted.

The reviews at this stage are the System Requirements Review (SRR), Integrated Baseline Review (IBR), and Technology Readiness Assessment (TRA). The SRR checks progress in defining the technical requirements, and the balance and completeness of the configuration. IBR is a business focused meeting and is concerned with the management effort. They can be repeated, and if lucky you can avoid them. The TRA checks the maturity of the critical technologies in a metrics based approach. As with the last phase, most of the local companies have done this before assigning the tasks to our teams. Completion of this phase is milestone B.

3.1.4 System Development and Demonstration Phase

This is the main engineering design and analysis stage. Some relevant IEEE Software Document Definitions for this phase are:

SQAP Software Quality Assurance Plan, IEEE 730

SCMP Software Configuration Management Plan, IEEE 828

STD Software Test Documentation, IEEE 829

SRS Software Requirements Specification, IEEE 830

SVVP Software Validation & Verification Plan, IEEE 1012

SDD Software Design Description, IEEE 1016

SPMP Software Project Management Plan, IEEE 1058

GDSRS Guide for Developing System Requirements Specifications, IEEE 1233

Functional Design

The specifications from the earlier phases are integrated with user input to make a set of functional requirements. Functional requirements specify what the system does rather than how it does it. Functional specifications also include user interactions (use cases). Together with background information, constraints (such as integration requirements, maintenance requirements, and safety requirements), and verification procedures and tests, these constitute the Functional Requirements Specification (FRS). The FRS goes by many names, but the key idea is it describes the required behavior of the engineering system without description of how it accomplishes the behavior. The FRS is designed to separate the needs from how the needs are met, and thus is the ideal tool to clarify the understanding of the problem before beginning the design. Use Case diagrams are designed to carry functional requirements, and as such are typically part of an FRS. The document should contain the acceptance procedures to show whether the design satisfies the requirements. When the FRS is ready, System Functional Review (SFR) is convened to examine the design and functional requirements to ensure they will meet the customer requirements and needs specified in earlier stages.

Preliminary Design

With a FRS in place, a preliminary design of the sub-systems may take place. Typically this is done at a high level to ensure that the performance specs can be met, and to make costing more precise. When the preliminary design and analysis is done a Preliminary Design Review (PDR) is conducted. The PDR is supposed to allow many eyes to look over the design before proceeding. Small flaws, misunderstandings, or mistakes become much more expensive later. This is a gate to stop flawed design ideas from proceeding. Many of our local companies fuse the functional design and preliminary design and just hold a PDR (skipping the SFR).

Detailed Design

The FRS and preliminary design is now expanded into a full Product Specification (PS) or Software Requirement Specification (SRS). The PS or SRS adds non-functional requirements to constrain the implementation, such as data structures, techniques, and algorithms to be used. The IEEE Prototype SRS outline from IEEE-STD-830-1998 is

```
Table of Contents
1. Introduction
```

```
   1.1 Purpose
   1.2 Scope
   1.3 Definitions, acronyms, and abbreviations
   1.4 References
   1.5 Overview
2. Overall description
   2.1 Product perspective
   2.2 Product functions
   2.3 User characteristics
   2.4 Constraints
   2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations
   of possible specific requirements. See also Annex A for several
   different ways of organizing this section of the SRS.)
Appendixes
Index
```

The IEEE has a good overview with a nice example of how to format, and CSUSB has a template online for master's projects.

When the PS or SRS is ready, a Critical Design Review (CDR) is held. This is the final review before The testing portion begins. The design is analyzed to see if it will meet the requirements (functional, safety, performance, etc.). The finances, production schedules, and risk assessments are updated. Approval means testing is to begin.

3.1.5 System Demonstration

The system demonstration sub-phase consists of setting the tests in accordance with the test master plan and PS/SRS, and low-level initial productions to verify manufacturability and quality of the resulting product. I will lightly cover this area because many small companies do this as part of the CDR. The main reviews of this part are the Test Readiness Review (TRR), System Verification Review (SVR), and Production Readiness Review (PRR). TRR assesses the test objectives, methods, procedures, scope, and safety. The TRR also verifies the test resources have been allocated. The SVR looks at the results of the tests and sees if low-rate initial production can begin. SVR is typically an audit trail of the CDR. The PRR looks at the low-rate initial production and the risk and cost factors to determine if full production can begin. At which point milestone C has been reached.

3.1.6 Production and Deployment Phase

Mostly engineering is support in this phase, except in software, where this is where coding happens. Testing and subtle improvements and fixes to attain the goals and meet customer needs is often required. Engineering change orders are the major tool, as are redline prints (marked up drawings or documents that must be entered into the document database). Sometimes an Operational Test Readiness Review (OTRR) is conducted to see if a mission critical or life critical product is ready for field testing. An OTRR can also take the form of Alpha testing. A Physical Configuration Audit (PCA) could also be done to decide if full rate production should begin. This determines if all goals have been met, and roughly corresponds to beta testing in software. When this is done the product is clear for full use.

3.1.7 Operations and Support Phase

Support for the product so it achieves its full use and duty. When the duty period is over safe disposal is necessary, which can involve ensuring no hazardous materials are dumped for physical products and/or that sensitive data is properly handled for computer systems and software.

3.2 Agile Methods

Agile methods are designed to switch from large, formal processes that are run top-down to small, flexible ones that are run bottom-up. While hierarchical (top-down) methods have historic dominance, Agile techniques dominate now, and their flexibility and creativity are likely to keep them on top for a while. They are far from free of criticism, and though I personally like them, I am more of a hybrid techniques person.

3.2.1 SCRUM

daily scrum
sprints
scrum master

3.2.2 Extreme Programming

small, frequent deliverables
have company rep close in the design process
reduce documentation, write code clearly don't document

3.2.3 Ideas Worth Stealing

Gantt Charts
Burndown Charts
responsibility/task wall chart

3.3 Quality Programs

In the 80's and 90's there was Total Quality Management, in the 2000's came Six Sigma design, what their goal is to do the right thing the right way. This involves a lot of work and is not as easy as I make it sound. In this chapter we will go over many of the pillars of this process.

Six goals of quality programs:

1. Customer focus (internal and external, this is the most popular emphasis)
2. Leadership and individual responsibility (let them do their job)
3. Integrity (quality as a way of life)
4. Enhance communication (avoid silly mistakes)
5. Recognize and reward employees (read find the best and keep them)
6. Streamline processes (save money, the real goal)

The goals are achieved by:

1. Customer focus (Kind of circular don't you think, you could also think of this a statement of the obvious, which sadly most people don't do.)
2. Involvement (The entire organization must support the goals, as a spoiler is deadly.)
3. Metrics (You have to have measurements to improve something)
4. Support (Budget, planning, etc.)
5. Continuous improvement (None of us is perfect, so try to get better at everything in a deliberate fashion.)

Quality programs are all about doing the right things, the right ways. You will incur certain necessary costs to do things right, but you will avoid many other costs and keep happy customers. Most big companies have some sort of quality program, but varying degrees of implementation. Most small companies have no formal system but are more naturally connected to customer needs. In the end, you need to get the customer what is required for the best price and profit.

The moral to me is

> *Doing it right costs, doing it wrong costs more.*

Before we examine the five ways we achieve our goals, we need to scope our discussion to the processes we will be working on. We can break down all we do, including our engineering designs (and the subsequent manufacturing) into a series of processes. Roughly, a process is a series of related tasks that produce a product. The product need not be physical or the final item. It is often useful to note that processes act on an input to produce a desired output according to a specification. Processes have:

boundaries Identify who are the actors and what they must do (responsibilities).

suppliers Produce inputs to according to the process's requirements.

customers Receive outputs and specify requirements.

The best processes are:

Effective Does it work? That is does it produce an output that meets the requirements.

Efficient Does it produce the output at the lowest cost?

Controlled The process is documented, and has metrics that are used to update and improve the process.

Adaptable Does it have built-in mechanisms that allow it to meet new requirements?

Processes must be analyzed and made to be effective, efficient, controlled, and adaptable. Common symptoms that indicate a problem are: customer complaints/dissatisfaction, returns, redos, unresolved issues, excessive workload/overtime, missed deadlines, budget difficulties, bad morale, turnover, productivity drops, constantly changing requirements. Note that most processes aren't going to meet the ideal. You need to find the most problematic and fix it, then repeat.

3.4 Customer Focus

Know your customer, and how your customer uses your product. An ideal customer-supplier relationship is aligned, that is, the supplier's abilities match the customer's requirements. To ensure alignment you need to ask what the customer needs, and what will they use it for. Particularly ask if there are any gaps between what is received and what is needed, often this opens up new opportunities. Make sure you keep written notes of conversations, and have people ok your summaries and requirement specifications. Be inclusive in the meetings, by which I mean you should actively invite those who have an interest in the product/service to be supplied. Make sure you have though things through before you meet so you don't look uninformed or ill-prepared. Think about what they will say and plan questions to elicit their inputs and concerns.

One way of thinking about customer focus and establishing requirements is the acronym PRIDE:

Product/service Does it meet the customer's requirements, needs, and wants? Note they are different. Did you meet them in a pleasing way?

Relationship Is their trust? How about a casual friendship?

Integrity Can you meet the requirements? What will you do if there is a problem?

Delivery Does it arrive on time and budget? Is it usable?

Expense Is the customer happy with the value? Is the price competitive?

3.5 Continuous Improvement

Have you ever heard this saying?

> *If it ain't broke, don't fix it.*

There are certainly times when this is the right course of action[1], but often it is not true. If man followed that saying in all areas, we would be hunting for dinner with a pointed stick[2]. Improvement is risky but the rewards are immeasurable. Improvement is what engineering is all about.

A popular rule is the 1-10-100 rule. Basically it says if you catch a thing in design it costs 1 buck, if in manufacturing it costs 10, and if your customer finds it it will cost you a 100. The point of this rule is to underscore the need for improvement. Four techniques that help improve are active listening, visualization, the why technique, and contingency diagrams.

1. Active listening means paying close attention to what everyone is saying. This involves having your mind and ears working together. Often great ideas or major dangers are brought up in discussions, but don't get noticed or are ignored. If we work hard at listening to others and thinking of their point and perspective then we will all benefit.

2. Visualization is the process where we imagine what we are making, how it will be used and misused, and how it benefits others. Think over the product. Put yourself in the place of needing it and using it. Many people do not consider what they are working on and thus make junk, don't join their ranks.

3. The why technique helps find causes by repeatedly asking why something happens. It is simple, but surprisingly effective, which is a good combination. It is particularly useful in tracking down a failure, but can be used proactively, particularly in trying to figure out how to make something work or to come up with new features.

4. Contingency diagrams[3] are a simple drawing in which a problem is listed in a bubble and the contributing factors listed on arrows pointing to the bubble. For each arrow, one or more solutions are developed. Like the why technique, it is particularly useful in tracking down a failure, but can also be used proactively, in trying to figure out how to make something work or to come up with new features.

Continuous improvement also applies to you. Your skills will degenerate and your mind grow weaker if you don't work at improving them. Do mind challenges like crosswords,

[1] Most quality people would shoot me for that, but it is true sometimes. Knowing when to leave it alone and when to improve it is an art you will develop over your life. You can't always trust little sayings, or always reject them. You must learn to think and assess on your own the best ways to handle things. Sometimes that means making and following rules. Sometimes you will have to break the rules. Great engineers and engineering managers know how and when to do that.

[2] Come to think of it, this sounds too much like a fun vacation or a cable survival show. Hopefully you get the point anyway.

[3] Personally I dislike the name, and have never thought it was very descriptive, but I am stuck with it.

sodoku, or brain teasers. Read popularizations of math, physics, computer science, engineering, philosophy, or history. I have a list of suggested books on my r2labs.org website if you need suggestions. Join and be active in professional societies like IEEE, SIAM, and ACM. Read the latest advancements in science and technology. Take certification courses. In short, keep getting better.

3.6 Manager's Side

Managers are responsible for the guiding and empowering of their employees to achieve success for the company. Many managers sadly see their job as keeping costs down, and thus they avoid improving things. One improvement frequently dumped for cost reasons is training. I don't know who first said it, but it is very true,

> *Training is a non-recurring overhead expense;*
> *Ignorance is a recurring direct charge.*

As a manager there are several things you can do to help your team improve.

- Give your group the big picture. To do a truly great job, and constantly improve, they need to know what is really going on. Many improvements happen when people realize how others use their work.
- Actively solicit new ideas. Most people are afraid to say things and contribute. You must make them feel comfortable with you and then seek their input and ideas. Avoid defensiveness and finger pointing (two common actions) and replace them with a desire to learn from mistakes and an open sharing of knowledge.
- Encourage your team to communicate directly with their customer. Make each person responsible for finding out how their work is used and empower them to do so. Bad managers try to control information flow. If you have an employee who is not able to communicate with customers then you have three choices: train them, pair them with a more skillful person, or if all else fails fire them.
- Have the team flowchart their work process and look for ways to improve it. Note that this is really helpful, but if you do it when a major task is due, you are crazy. Quality takes time and effort, no matter what people may tell you.
- Model high standards in your work and reward your team for their work. Encourage and reward when they succeed, you never lose by doing this as it reflects well on you.
- Seek and grow the best people on your team. Make the environment supportive, open, and friendly. People want to work in these type of places, and will often work harder for less if they are working for and with friends. This is not a license to take advantage of people, rather it means that if you all help each other, then you will all succeed together and be happy.

One thing you should keep to a minimum is meetings. Meetings are necessary to keep people informed, but they block real work and create frustration in employees (particularly the good ones). Make sure meetings are planed and stay on track.

Often you will need to guide your team to solve a problem. Here is a four step method I learned at Grumman Aerospace (now Northrop-Grumman).

1. Focus
   (a) List of problems (brainstorm)
   (b) Select one problem (selection grid)
   (c) Verify/Define problem (impact analysis)

   Output: problem statement

2. Analyze
   (a) Identify needed information (checklist)
   (b) Collect data (sampling, surveys, literature)
   (c) Find major factors (statistics, flowchart, fishbone, why)

   Output: baseline data, and list of contributors

3. Develop
   (a) List possible solutions (brainstorming, literature)
   (b) Select best solution (cost-benefit analysis)
   (c) Develop plan (task list, gantt chart, procedure)

   Output: plan

4. Execute
   (a) Buy-in (presentations, reports/papers, individual discussions)
   (b) Implementation
   (c) Monitoring and tuning

   Output: metrics

3.7 Configuration Management

In truth, configuration management is an oversight process that uses systems engineering. Configuration management is the use of business, engineering, manufacturing, and logistics practices to maintain consistency of the product. This entails the tracking and version control of all documents, drawings, code, reviews, and data associated with a project. This is particularly important in ensuring the latest modifications and fixes get used promptly, and any shipped items get the needed retrofits. It also includes the oversight of all processes involved in the design, production, testing, and maintenance of the project, such as material procurement, tooling, and training of personnel. This is a large topic, and is usually covered in Software Engineering. We will cover only selected portions, which are of particular interest to us at the start of your education. For more information on configuration management see [1, 5, 7, 10].

3.7.1 Document Control

Git and the web site github are my preferred way to track documents and software. Git is very compatible with Agile techniques, and seeks to empower the individual while making an integrated work-flow to keep a group moving together.

Chapter 4

Engineering Research

Why are we doing research to design? It is always a good idea to start knowing why you want to do something. The justification is also a major reason. We are about to begin an engineering design, which is an expensive process. We need to make sure we do the right design, and that there was not a better solution out there.

4.1 Know Who is Involved

The first thing we want to do is identify our stakeholders. Stakeholders are anyone who is involved in or is affected by the design. The basic list is:

- client
- user
- anyone affected
- management
- manufacturing
- purchasing

4.2 Know the Project

Many people design great solutions to the wrong problem. Often what the client asks for is the wrong solution, because of an assumed solution. A good engineer needs to be entrepreneurial in finding the real problem, and thus designing a solution that really works. You need to find out:

- problem or situation that motivated the project

- need that was not met
- who will use it and how it will be used
- regulations
- any past attempts or current work that has been done
- existing (similar) solutions from other sources
- time constraints
- resources available

4.3 Be SMART

Ultimately you want to take your research and make SMART goals.

- specific
- measurable
- achievable
- relevant
- time-bound

4.4 Technology Readiness Level

Technology Rediness Level (TRL) is a measure of the maturity of a technology in the sense that it can be readily used in a design. The scale runs for 1-9, with 1 meaning it is an idea people are thinking about to 9 meaning heavily used and tested.

4.5 How to Brainstorm

This sounds easy right. Get a bunch of people together and think up solutions. Sadly, these often fail because of poor implementation. I strongly suggest you follow all of the following:

1. Have one person write the ideas on the board, and another document for later use in a notebook.
2. No criticism or analysis can be done. Not even a comment of "that is the same thing" as this will disturb the process and scare people from contributing. Things that come up repeatedly might be more important, and variants sometimes have important improvements.

3. Each person gets one contribution per turn, and turns go in sequence. Give people paper to jot down ideas so they don't forget. If you don't do this you run the risk of cutting some people out. People can say pass, but should be encouraged to say what they think, even if it is a repeat.

4.6 Risk

4.6.1 Identification

We want to identify what can go wrong as early as possible. This step is constantly going.

4.6.2 Analysis

Performance Consequence			Probability of Consequence				
Technical	Schedule	Cost	$p \leq 0.1$	$< p \leq 0.4$	$< p \leq 0.6$	$< p \leq 0.9$	$< p$
none or minimal	none or minimal	none or minimal	G	G	G	G	G
minor perf. loss, no program impact	able to meet key dates	+1% to +0%	G	G	G	Y	Y
moderate perf. loss, limited program impact	use up float, no critical path slips	+5% to +1%	G	G	Y	Y	R
significant perf. loss, major program impact	critical path slips	+10% to +5%	G	Y	Y	R	R
severe perf. loss, jeopardize program	milestones missed by multiple months	+10% or more	Y	Y	R	R	R

4.6.3 Mitigation Planning

Identify the root cause if it exists, and the necessary/enabling and sufficient/primary causes. Eliminate causes or consequences if possible, control/reduce them if they can't be eliminated. Transfer risks to more tolerant systems.

4.6.4 Implementation

Execution of plan, including budget, management, and technical efforts.

4.6.5 Tracking

Monitor success to see if errors happened at any stage and modify accordingly.

Chapter 5

Engineering Economics

Engineers have to consider the cost of the items designed, thus we must know some economics. This is an introduction to engineering economics. Those regularly dealing with costing are encouraged to study other references for a deeper coverage. This should get you started though.

5.1 Financial Preliminaries

Some standard variables I will use are:

P Present sum of money, sometimes called principle though this does not always make sense.

F Future sum of money, at n periods in the future.

A Annuity, or regular payment of the same amount.

G Geometric payment series.

i Interest rate per compounding period, usually a year.

n Number of interest periods.

r Nominal interest rate per year, does not consider effect of subperiod compounding.

m Number of compounding subperiods.

5.1.1 Simple Interest

We want to study finance, and to do so we must look at interest. The most basic kind of interest is simple interest. While not used in the US, simple interest is used in some countries in Africa. Let's get a few terms down first.

Principle (P) the amount borrowed or initial balance.

Future worth (F) worth of the investment at that interest rate.

Interest rate (i) the percent of the principle charged for the loan.

The interest charged each year is the same in simple interest, and can be found by multiplying the interest rate times the amount owed, Pi. Simple interest is an example of arithmetic growth. The series that shows how much must be paid for a simple interest loan of P at rate i for n years is

$$\begin{aligned} F &= P + Pi + Pi + \cdots + Pi \\ &= P + nPi \\ &= P(1 + ni) \end{aligned} \tag{5.1}$$

5.1.2 Compound Interest

In the US we do not use simple interest, but rather compound interest. Before we get into the formal ideas, let's see why it is called compound interest.

Say you put P dollars in a savings account at an interest rate of i. We will assume the interest is calculated and added to your account only once a year. The interest earned at the end of the year is Pi, just as in simple interest. After 1 year you have $P(1+i)$ dollars. If you leave the money in the account then the next year you will have a new initial amount of $P(1+i)$, so the interest paid will be $(P(1+i))i$. The total money you have will be $P(1+i)(1+i) = P(1+i)^2$. As this continues, the money you have at the end of n years is

$$F = P(1+i)^n. \tag{5.2}$$

The interest builds upon itself to help you, or compound your gain.

We do not have to compound (or add) the interest just once a year, you can compound a number of times a year. Say we compounded interest m times per year, at the end of a year we would have $P(1+i)^m$ dollars, right?. Interestingly (pardon the pun) no bank uses the same i in yearly interest compounding as in more frequent compounding. The interest used when interest is compounded more than once a year is $r = \frac{i}{m}$. If a bank lists that an account pays a nominal (APR) of $i = rm$ that is compounded m times per year, the rate used in the compounding equation is r. The formula reads $P(1+\frac{i}{m})^m = P(1+r)^m$. If we leave our money in for two years we would have $P(1+\frac{i}{m})^{2m} = P(1+r)^{2m}$. Sometimes instead of listing the nominal interest (APR), a bank will list the effective rate (APY). This comes from noticing that $(1+r)^m > 1$, so we can write it as $(1+r)^m = 1 + r_e$. We then solve for r_e to find that $r_e = (1+r)^m - 1$. You would have this actual interest rate if the account were compounded annually. After n years, the balance will have grown to

$$F = P(1+\frac{i}{m})^{nm} \tag{5.3}$$

$$= P(1+r)^{nm} \tag{5.4}$$

$$= P(1+r_e)^n. \tag{5.5}$$

This series is a geometric series. This is a commonly calculated equation, so we will do a little bit of work to get it into an even more useful form.

$$F = P(1+r)^k$$
$$\frac{F}{P} = (1+r)^k \tag{5.6}$$
$$= (F/P,\ r,\ k) \tag{5.7}$$

The final equation (eq. 5.7) is called the **single payment compound amount factor**, and is tabulated in many places for different interest rates. This allows for easy calculation. It is also easy to code the formula for programs.

Easy Estimation

One challenge with compounded interest is estimating what something is worth in 20 years at some interest rate. There is an easy way to estimate this value.

> **Rule of 69** A sum invested at an interest rate, r, will double every k years where, $k = \frac{69}{100r}$.

For example, at 3% interest, an investment doubles every $\frac{69}{3} = 23$ years. Try it $(1.03)^{23} \approx 1.97$, a very good approximation. This can be used to rapidly estimate values.

How does it work? Magic of course. Just kidding. Here is the proof

$$2 = (1+r)^t \tag{5.8}$$
$$\ln(2) = t\ln(1+r) \tag{5.9}$$
$$t = \frac{\ln(2)}{\ln(1+r)} \tag{5.10}$$
$$t \approx \frac{0.69}{r} \tag{5.11}$$

Continuous Compounding

You might have noticed something when we looked at multiple compoundings per year. You might have noticed that the more frequent the compoundings, the effective interest rates (APY) increased but not as much each time. Let's look at this trend more closely, see Figure 5.1.

The final term is a special number in mathematics called e. There are many similarities between the constants e and π. Here are a few.

1. Both arise naturally in calculations. Circles, curves, and angles give rise to π, while compounding and calculus operations on exponents give rise to e.

2. Both numbers are not rational, which means they cannot be expressed as the ratio of integers.

Figure 5.1: Continuous Compounding (digits rounded to 8 for display purposes)

n	$1 + APY$
1.	2.
2.	2.25
3.	2.3703704
4.	2.4414062
5.	2.48832
10.	2.5937425
20.	2.6532977
30.	2.6743188
40.	2.6850638
50.	2.691588
100.	2.7048138
200.	2.7115171
300.	2.7137652
400.	2.7148917
500.	2.7155685
1000.	2.7169239
Inf	2.7182818

3. Both numbers are not algebraic and thus they are transcendental. An algebraic number is one which can be expressed as the root of a polynomial with integer coefficients.

4. Both have no repeating patterns in their decimal expansion. This is a result of being irrational.

5.1.3 Fun with Numbers

I thought I would take this time to give you a little taste of Number Theory. This is the type of thing which mathematicians find fun. Lets look a few of them.

First, consider a repeating decimal like 0.345345345 We would like to know how to express this as a fraction. We know 345/1000 gives us 0.345, which is a little smaller than we want, so we make the denominator a little smaller. We try 345/999 and we find that indeed we get the desired result. Try the following

7/9	75/99	752/999	7528/9999
5/9	50/99	504/999	5046/9999
1/9	12/99	123/999	1234/9999

Why does this work? Consider the following.

Let some number sequence of $n + 1$ digits be given by $d_0 d_1 \cdots d_n$. To make it easy to follow note that $n + 1$ digits of all 9's is specified by $9_0 9_1 \cdots 9_n$. Finally we will look at the

Table 5.1: Digit recursion.

x	explanation
$0.d_0d_1\cdots d_n$	The first $n+1$ digits must be from the fixed number since the shift is large enough so there is no overlap.
$0.d_0d_1\cdots d_n\ d_0d_1\cdots d_n$	Since we know the first $n+1$ digits, we know from the shift term that the first $n+1$ digits are copied and shifted to the second $n+1$ digits.
$0.d_0d_1\cdots d_n\ d_0d_1\cdots d_n\ d_0d_1\cdots d_n$	Since we know the second $n+1$ digits, we know from the shift term that the second $n+1$ digits are copied and shifted to the third $n+1$ digits.
$\vdots$	The process repeats, generating the pattern.

ratio,

$$
\begin{aligned}
x &= \frac{d_0d_1\cdots d_n}{9_09_1\cdots 9_n}\\
&= \frac{d_0d_1\cdots d_n}{10^{n+1}-1}\\
x(10^{n+1}-1) &= d_0d_1\cdots d_n\\
x10^{n+1} &= d_0d_1\cdots d_n + x\\
x &= \frac{d_0d_1\cdots d_n}{10^{n+1}} + x\frac{1}{10^{n+1}}\\
&= 0.d_0d_1\cdots d_n + x\frac{1}{10^{n+1}}
\end{aligned}
$$

Think about what this equation means. The ratio, x, is equal to a fixed length ($n+1$ digits) decimal number and a shifted (by $n+1$) copy of itself. Since the shift is equal to the size of the fixed number, they don't overlap. We can then see that this can be handled recursively, see Table 5.1.

Note the term in square brackets is repeat of the term on the on line two so we have for 345/999, that it is 0.345(1.001001…) or 0.345345345… We have seen that this works in general. Try 9/9. We should see 0.99999 from what we have shown, but we see 1! Why? Well, 1=0.99999…! Here is an easy way to see this.

$$
\begin{aligned}
y &= 0.999\cdots\\
10y &= 9.999\cdots\\
10y-y &= 9.999\cdots - 0.999\cdots\\
9y &= 9\\
y &= 1
\end{aligned}
$$

An amazing result! Anyway, I hope this little diversion sparks an interest in number theory.

5.1.4 Amortization and Savings

Up till now we have looked at a single starting value and what it will be worth (or cost). Now we will consider amortization and regular payments. Say we are going to make regular payments of A for m periods per year and n years. Further, say you will get $r = \frac{i}{m}$ as an interest rate. We can treat this as a bunch of compounded equations that are added together.

$$\begin{aligned} F &= A(1+r)^{nm-1} + A(1+r)^{nm-2} + \\ &\qquad \ldots + A(1+r) + A \\ &= \sum_{k=0}^{nm-1} A(1+r)^k \\ &= A \sum_{k=0}^{nm-1} (1+r)^k \end{aligned} \tag{5.12}$$

Note that this is nice looking series, so we might suspect there is an easier way of summing. Consider the series

$$s_m(x) = \sum_{k=0}^{m-1} x^k, \tag{5.13}$$

then multiply by $\frac{(x-1)}{(x-1)}$ to obtain

$$\begin{aligned} s_m(x) &= \frac{x s_m(x) - s_m(x)}{x-1} \\ &= \frac{x\left(x^{m-1} + s_{m-1}(x)\right)}{x-1} \\ &\quad - \frac{(x s_{m-1}(x) + 1)}{x-1} \\ &= \frac{x^m + x s_{m-1}(x)}{x-1} \\ &\quad - \frac{x s_{m-1}(x) - 1}{x-1} \\ &= \frac{x^m - 1}{x-1}. \end{aligned} \tag{5.14}$$

We note that for us $x = 1 + r$ so we have

$$\begin{aligned}
F &= A \sum_{k=0}^{nm-1} (1+r)^k \\
&= A \frac{(1+r)^{nm} - 1}{(1+r) - 1} \\
&= A \frac{(1+r)^{nm} - 1}{r} \\
\frac{F}{A} &= \frac{(1+r)^{nm} - 1}{r} && (5.15) \\
&= \left(\frac{F}{A},\ r,\ nm\right). && (5.16)
\end{aligned}$$

This is called the amortization or savings formula.

When paying off a mortgage or loan, the future value of the amortization (your payment) and the future value of the initial amount of the loan are set equal. Assuming multiple compoundings we have

$$\begin{aligned}
P(1+r)^{nm} &= A \frac{(1+r)^{nm} - 1}{r} \\
A &= \frac{P(1+r)^{nm} r}{(1+r)^{nm} - 1} \\
\frac{A}{P} &= \frac{r(1+r)^{nm}}{(1+r)^{nm} - 1} \\
&= \left(\frac{A}{P},\ r,\ nm\right). && (5.17)
\end{aligned}$$

The final equation (eq. 5.17) is called the **Capital Recovery Factor**, and it is tabulated for various values of interest and compounding periods. Notice that this is the formula used for loans (loan value is P, monthly payment is A, and the number of monthly payments is nm). The reciprocal,

$$\begin{aligned}
\frac{P}{A} &= \frac{(1+r)^{nm} - 1}{r(1+r)^{nm}} && (5.18) \\
&= \left(\frac{P}{A},\ r,\ nm\right), && (5.19)
\end{aligned}$$

is also an important equation, and is called the **Present Worth Factor, Uniform Series**.

5.1.5 Arithmetic Gradients

Compound amount, arithmetic gradient

$$\frac{F}{G} = \left(\frac{(1+r)^k - (1+rk)}{r^2}\right) \tag{5.20}$$
$$= \left(\frac{F}{G},\ r,\ k\right) \tag{5.21}$$

Present worth factor, arithmetic gradient

$$\frac{P}{G} = \left(\frac{(1+r)^k - (1+rk)}{r^2(1+r)^k}\right) \tag{5.22}$$
$$= \left(\frac{P}{G},\ r,\ k\right) \tag{5.23}$$

Annuity factor, arithmetic gradient

$$\frac{A}{G} = \left(\frac{(1+r)^k - (1+rk)}{r(1+r)^k - r}\right) \tag{5.24}$$
$$= \left(\frac{A}{G},\ r,\ k\right) \tag{5.25}$$

5.1.6 Conversion

Let's summarize our conversion factors. Note that often we have combinations of terms to describe some situation, but with our conversion table we can handle this.

	From			
To	P	F	A	G
P	1	$(1+r)^{-k}$	$\frac{(1+r)^k-1}{r(1+r)^k}$	$\frac{(1+r)^k-(1+rk)}{r^2(1+r)^k}$
F	$(1+r)^k$	1	$\frac{(1+r)^k-1}{r}$	$\frac{(1+r)^k-(1+rk)}{r^2}$
A	$\frac{r(1+r)^k}{(1+r)^k-1}$	$\frac{r}{(1+r)^k-1}$	1	$\frac{(1+r)^k-(1+rk)}{r(1+r)^k-r}$
G	$\frac{r^2(1+r)^k}{(1+r)^k-(1+rk)}$	$\frac{r^2}{(1+r)^k-(1+rk)}$	$\frac{r(1+r)^k-r}{(1+r)^k-(1+rk)}$	1

As odd as it might sound, we often prefer to use just the labels, because we can tabulate or do function calls to get the solution, which tends to be less error prone than doing arithmetic. See Appendix E for sample compound interest table factors.

	From			
To	P	F	A	G
P	1	$(P/F, r, k)$	$(P/A, r, k)$	$(P/G, r, k)$
F	$(F/P, r, k)$	1	$(F/A, r, k)$	$(F/G, r, k)$
A	$(A/P, r, k)$	$(A/F, r, k)$	1	$(A/G, r, k)$
G	$(G/P, r, k)$	$(G/F, r, k)$	$(G/A, r, k)$	1

5.2 Returns and Comparison Methods

5.2.1 Equivalent Uniform Annual Cost(EUAC)

$$\begin{aligned} EUAC &= \left(\frac{A}{P}, r, k\right) P \\ &\quad - \left(\frac{A}{F}, r, k\right) S \end{aligned} \tag{5.26}$$

where P is the cost of the item, and S is the salvage value (what can you sell it for when you are done).

5.2.2 Equivalent Uniform Annual Benefit(EUAB)

$$EUAB = A - \left(\frac{A}{P}, r, k\right) P \tag{5.27}$$

where P is the cost of the investment, and A is the periodic return.

5.2.3 Net Present Worth(NPW)

Net present worth (NPW) is defined by

$$NPW = \left(\frac{P}{A}, r, k\right) A - P \tag{5.28}$$

where P is the cost of the investment, and A is the periodic return.

5.2.4 Rate of Return(ROR)

The rate of return is the interest rate, which makes the net present worth equal zero. The rate of return is thus:

$$\begin{aligned} ROR &= r \\ &\quad s.t.\ NPW = 0, \end{aligned} \tag{5.29}$$

where $s.t.$ means "such that"[1]. ROR is usually evaluated with respect to a minimum attractive rate of return (MARR), which is selected to reflect current options and relative risk of the investment. MARR can be selected to be a safe investment's ROR (such as a savings account or secure bonds).

[1] Such that means that there is a constraint that must be fulfilled. It is frequently used in constrained minimization, but can be used - as it is here - to specify a one parameter system from which any solution will do.

5.3 Depreciation

We have seen how things grow. We also need to see how things lose value. Major assets (homes, cars, computers, etc.) do not retain their value. They get used and thus worth less to others, some more than others. For instance, cars tend to lose 25% of their value a year until they are in the $1000 range where they tend to hover (unless they are a classic). Houses on the other hand tend not to lose a lot of value and actually start gaining value after about 5-10 years of age. Depreciation works just like interest, except the rate is negative.

Consider a new car that costs $20,000 and loses 25% of its value each year. How much is it worth in 1, 2, 5, 10 years?

Years	Formula	Value
0	$20000(1-0.25)^0$	$20,000
1	$20000(1-0.25)^1$	$15,000
2	$20000(1-0.25)^2$	$11,250
5	$20000(1-0.25)^5$	$4,746
10	$20000(1-0.25)^{10}$	$1,126

People often find it difficult to calculate this, so other methods are often used. Some of the most common are below.

5.3.1 Straight Line Depreciation

$$D_k = \frac{C-S}{n} \tag{5.30}$$

where D_k is the depreciation in year k, C is the cost of the item, S is the salvage value (what you get for selling it at the end, and n is the number of years till you salvage it.

5.3.2 Accelerated Cost Recovery System (ACRS)

$$D_k = W_j(C-S) \tag{5.31}$$

where D_k is the depreciation in year k, C is the cost of the item, S is the salvage value (what you get for selling it at the end, and W_j is from Table 5.2.

5.4 Resources

5.4.1 Non-Renewable Resources

If you have a sum of money, a quantity of oil, etc., that you cannot replace then it is a non-renewable resource. There are two ways you can use it (in practical situation, theoretically you can use it in infinitely many ways). Our primary interest is when the resource will be used up.

Table 5.2: ACRS Weights

	Recovery Period			
Year	3	5	7	10
1	.333	.200	.143	.100
2	.445	.320	.245	.180
3	.148	.192	.175	.144
4	.074	.115	.125	.115
5		.115	.089	.092
6		.058	.089	.074
7			.089	.066
8			.045	.066
9				.065
10				.065
11				.033

The first is static consumption. In it you use the same amount, A, every period. The resource will last

$$t_s = \frac{P}{A} \tag{5.32}$$

The second type is exponential consumption. In it you start by using A the first period, and then you use A(1+r) the next period, and so on. The equation ends up looking like the savings formula (though we are withdrawing not depositing). In any case the formula is

$$\begin{aligned} P &= A + A(1+r) + \cdots + A(1+r)^{t_e-1} \\ &= A\frac{(1+r)^{t_e}-1}{r} \end{aligned}$$

We can then solve for t_e, the time when it is all gone.

$$\frac{P}{A}r + 1 = (1+r)^{t_e} \tag{5.33}$$

$$\ln\left(\frac{P}{A}r + 1\right) = t_e \ln(1+r) \tag{5.34}$$

$$t_e = \frac{\ln\left(\frac{P}{A}r+1\right)}{\ln(1+r)} \tag{5.35}$$

5.4.2 Renewable Resources

If you have a sum of money, a group of animals, an amount of plants, etc., that can grow, you have a renewable resource. In this case it makes sense to live off the surplus amount above that which is needed to maintain the quantity at current levels. For money you live off the interest.

If we ignore inflation the amount we get in one year is given by

$$
\begin{aligned}
F &= P(1+r) && (5.36)\\
&= P + rP. && (5.37)
\end{aligned}
$$

We want to keep an amount P left over (for the following year) so the net gain (what we can live off) is

$$
\begin{aligned}
A &= F - P && (5.38)\\
&= P + rP - P && (5.39)\\
&= rP. && (5.40)
\end{aligned}
$$

We might want to consider inflation into this equation, so the money we have to spend each year remains of equal value. In this case the amount we need to keep from spending is $P(1+i)$, which gives us an amount to live off of

$$
\begin{aligned}
A &= F - P(1+i) && (5.41)\\
&= P + rP - P - iP && (5.42)\\
&= (r-i)P. && (5.43)
\end{aligned}
$$

Since interest rates are determined by inflation (usually r=i+1% to 3%). For instance it is reasonable to assume that if you live off $\frac{1}{50}$ of an amount you can do so forever.

5.5 Estimating Costs

5.5.1 Engineering Cost Estimate Method

A bottom-up method that consists of estimating the costs for each sub area (labor, materials, overhead, etc.). Very detailed and often quite accurate. Costs are assigned to each definable task or activity in the work breakdown structure (WBS).

5.5.2 Roundtable Cost Estimate Method

Experts representing every discipline/area of the production meet to discuss cost based on experience. Typically this includes engineering (hardware, software, mechanical), manufacturing (production, assembly), flight test/quality assurance, logistics & purchasing, accounting, contracts, and human resources.

5.5.3 Actual Costs Extrapolation Method

Calculate the cost from prototype costs. This is very accurate but can't always be done and is not available early in the process.

5.5.4 Analogy Cost Estimating Method

Draws an analogy to an existing product, and uses the costing of the existing product. Differences can be specified, and cost modifiers used. This is the simplest method, and its accuracy requires that good historical data be available, a proper analogy drawn, and differences properly noted. It is not only simple, it can also be fast, and usually provides a good rough order of magnitude.

5.5.5 Parametric/Statistical Estimation Method

Use a database of similar systems to generate estimates of the cost. This is essentially a multiple item analogy cost estimate. Often used early in the design process. In this method we exploit correlation of variables. When considering why variables have a high correlation/relation in the data (indicating a potential dependence), there are three possible reasons:

1. No relation - it is chance.
2. Related but not causal - neither one caused the other, but some relation exists between the variables.
    (a) All dependent on missing variable. For example let x and y be two variables which exhibit a relation. It is possible that both are dependent and there is a missing independent variable, say t, such that $x = f(t)$ and $y = g(t)$. One such case is that of the amount of sunlight in two different places on the earth. It being daylight in LA does not cause it to be night in Mumbai, though it will be the case due to a missing variable. The shape and rotation of the earth causes the day/night phenomena noted.

    (b) They might be co-dependent, as you often see in symbiotic relations among plants and animals. Thus $x = f(y)$ and $y = g(x)$ and we cannot tell which is independent (and there often isn't one).

    (c) The might be two independent factors, where a joint distribution is known thus constraining them, but not being dependent in a causal way. For example $z = f(x, y)$, if we know the function and values of z, it will constrain the x and y used to pick it.
3. Causal - one variable caused the other to take on the value, such as an unbalanced force causing an acceleration.

Often we can get away with a linear fit, which has the virtue of being easy to calculate. To see how good this is we can use several goodness of fit measures.

Residual The sum of all the errors of fit, measured only in the dependent variable's direction. Also called standard error or standard error of estimate (not to be confused with statistical error).

Coefficient of Variation of Residuals Useful in comparing distributions with widely varying stats. See Appendix D.

Correlation Coefficient The covariance of the two variables divided by the variables standard deviations, see Appendix D.

Determination Coefficient The square of the correlation coefficient.

Chapter 6

Engineering Ethics

Engineering ethics are really more professional codes of conduct than true ethical systems. The main difference is that codes of conduct specify conduct rather than specifying the way to evaluate a situation. Amongst professional codes of conduct, there are two general methods, specific and general. Specific codes of conduct are typified by the Association of Computer Machinist's (ACM) code of conduct, which is very detailed in what is to be done in different situations. On the other extreme is the Institute of Electrical and Electronic Engineer's (IEEE) code of conduct, which has only some general principles.

6.1 IEEE Code of Ethics

As per IEEE Bylaw I-104.14, membership in IEEE in any grade shall carry the obligation to abide by the IEEE Code of Ethics (IEEE Policy 7.8) as stated below.

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

- to accept responsibility in making decisions consistent with the safety, health and welfare of the public, and to disclose promptly factors that might endanger the public or the environment;
- to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;
- to be honest and realistic in stating claims or estimates based on available data;
- to reject bribery in all its forms;
- to improve the understanding of technology, its appropriate application, and potential consequences;

- to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;
- to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;
- to treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin;
- to avoid injuring others, their property, reputation, or employment by false or malicious action;
- to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

Approved by the IEEE Board of Directors
February 2006

6.2 ACM Code of Ethics and Professional Conduct

Adopted by ACM Council 10/16/92.

Preamble

Commitment to ethical professional conduct is expected of every member (voting members, associate members, and student members) of the Association for Computing Machinery (ACM).

This code, consisting of 24 imperatives formulated as statements of personal responsibility, identifies the elements of such a commitment. It contains many, but not all, issues professionals are likely to face. Section 1 outlines fundamental ethical considerations, while Section 2 addresses additional, more specific considerations of professional conduct. Statements in Section 3 pertain more specifically to individuals who have a leadership role, whether in the workplace or in a volunteer capacity such as with organizations like ACM. Principles involving compliance with this Code are given in Section 4.

The Code shall be supplemented by a set of Guidelines, which provide explanation to assist members in dealing with the various issues contained in the Code. It is expected that the Guidelines will be changed more frequently than the Code.

The Code and its supplemented Guidelines are intended to serve as a basis for ethical decision making in the conduct of professional work. Secondarily, they may serve as a basis for judging the merit of a formal complaint pertaining to violation of professional ethical standards.

It should be noted that although computing is not mentioned in the imperatives of Section 1, the Code is concerned with how these fundamental imperatives apply to one's conduct as a computing professional. These imperatives are expressed in a general form to emphasize that ethical principles which apply to computer ethics are derived from more general ethical principles.

It is understood that some words and phrases in a code of ethics are subject to varying interpretations, and that any ethical principle may conflict with other ethical principles in specific situations. Questions related to ethical conflicts can best be answered by thoughtful consideration of fundamental principles, rather than reliance on detailed regulations.

1. GENERAL MORAL IMPERATIVES.

As an ACM member I will ....

1.1 Contribute to society and human well-being.

This principle concerning the quality of life of all people affirms an obligation to protect fundamental human rights and to respect the diversity of all cultures. An essential aim of computing professionals is to minimize negative consequences of computing systems, including threats to health and safety. When designing or implementing systems, computing professionals must attempt to ensure that the products of their efforts will be used in socially responsible ways, will meet social needs, and will avoid harmful effects to health and welfare.

In addition to a safe social environment, human well-being includes a safe natural environment. Therefore, computing professionals who design and develop systems must be alert to, and make others aware of, any potential damage to the local or global environment.

1.2 Avoid harm to others.

"Harm" means injury or negative consequences, such as undesirable loss of information, loss of property, property damage, or unwanted environmental impacts. This principle prohibits use of computing technology in ways that result in harm to any of the following: users, the general public, employees, employers. Harmful actions include intentional destruction or modification of files and programs leading to serious loss of resources or unnecessary expenditure of human resources such as the time and effort required to purge systems of "computer viruses."

Well-intended actions, including those that accomplish assigned duties, may lead to harm unexpectedly. In such an event the responsible person or persons are obligated to undo or mitigate the negative consequences as much as possible. One way to avoid unintentional harm is to carefully consider potential impacts on all those affected by decisions made during design and implementation.

To minimize the possibility of indirectly harming others, computing professionals must minimize malfunctions by following generally accepted standards for system design and testing. Furthermore, it is often necessary to assess the social consequences of systems to project the likelihood of any serious harm to others. If system features are misrepresented to users, coworkers, or supervisors, the individual computing professional is responsible for any resulting injury.

In the work environment the computing professional has the additional obligation to report any signs of system dangers that might result in serious personal or social damage. If one's superiors do not act to curtail or mitigate such dangers, it may be necessary to "blow the whistle" to help correct the problem or reduce the risk. However, capricious or misguided reporting of violations can, itself, be harmful. Before reporting violations, all relevant aspects of the incident must be thoroughly assessed. In particular, the assessment of risk and responsibility must be credible. It is suggested that advice be sought from other computing professionals. See principle 2.5 regarding thorough evaluations.

1.3 Be honest and trustworthy.

Honesty is an essential component of trust. Without trust an organization cannot function effectively. The honest computing professional will not make deliberately false or deceptive claims about a system or system design, but will instead provide full disclosure of all pertinent system limitations and problems.

A computer professional has a duty to be honest about his or her own qualifications, and about any circumstances that might lead to conflicts of interest.

Membership in volunteer organizations such as ACM may at times place individuals in situations where their statements or actions could be interpreted as carrying the "weight" of a larger group of professionals. An ACM member will exercise care to not misrepresent ACM or positions and policies of ACM or any ACM units.

1.4 Be fair and take action not to discriminate.

The values of equality, tolerance, respect for others, and the principles of equal justice govern this imperative. Discrimination on the basis of race, sex, religion, age, disability, national origin, or other such factors is an explicit violation of ACM policy and will not be tolerated.

Inequities between different groups of people may result from the use or misuse of information and technology. In a fair society,all individuals would have equal opportunity to participate in, or benefit from, the use of computer resources regardless of race, sex, religion, age, disability, national origin or other such similar factors. However, these ideals do not justify unauthorized use of computer resources nor do they provide an adequate basis for violation of any other ethical imperatives of this code.

1.5 Honor property rights including copyrights and patent.

Violation of copyrights, patents, trade secrets and the terms of license agreements is prohibited by law in most circumstances. Even when software is not so protected, such violations are contrary to professional behavior. Copies of software should be made only with proper authorization. Unauthorized duplication of materials must not be condoned.

1.6 Give proper credit for intellectual property.

Computing professionals are obligated to protect the integrity of intellectual property. Specifically, one must not take credit for other's ideas or work, even in cases where the work has not been explicitly protected by copyright, patent, etc.

1.7 Respect the privacy of others.

Computing and communication technology enables the collection and exchange of personal information on a scale unprecedented in the history of civilization. Thus there is increased potential for violating the privacy of individuals and groups. It is the responsibility of professionals to maintain the privacy and integrity of data describing individuals. This includes taking precautions to ensure the accuracy of data, as well as protecting it from unauthorized access or accidental disclosure to inappropriate individuals. Furthermore, procedures must be established to allow individuals to review their records and correct inaccuracies.

This imperative implies that only the necessary amount of personal information be collected in a system, that retention and disposal periods for that information be clearly defined and enforced, and that personal information gathered for a specific purpose not be used for other purposes without consent of the individual(s). These principles apply to electronic communications, including electronic mail, and prohibit procedures that capture or monitor electronic user data, including messages,without the permission of users or bona fide authorization related to system operation and maintenance. User data observed during the normal duties of system operation and maintenance must be treated with strictest confidentiality, except in cases where it is evidence for the violation of law, organizational regulations, or this Code. In these cases, the nature or contents of that information must be disclosed only to proper authorities.

1.8 Honor confidentiality.

The principle of honesty extends to issues of confidentiality of information whenever one has made an explicit promise to honor confidentiality or, implicitly, when private information not directly related to the performance of one's duties becomes available. The ethical concern is to respect all obligations of confidentiality to employers, clients, and users unless discharged from such obligations by requirements of the law or other principles of this Code.

2. MORE SPECIFIC PROFESSIONAL RESPONSIBILITIES.

As an ACM computing professional I will ....

2.1 Strive to achieve the highest quality, effectiveness and dignity in both the process and products of professional work.

Excellence is perhaps the most important obligation of a professional. The computing professional must strive to achieve quality and to be cognizant of the serious negative consequences that may result from poor quality in a system.

2.2 Acquire and maintain professional competence.

Excellence depends on individuals who take responsibility for acquiring and maintaining professional competence. A professional must participate in setting standards for appropriate levels of competence, and strive to achieve those standards. Upgrading technical knowledge and competence can be achieved in several ways:doing independent study; attending seminars, conferences, or courses; and being involved in professional organizations.

2.3 Know and respect existing laws pertaining to professional work.

ACM members must obey existing local, state,province, national, and international laws unless there is a compelling ethical basis not to do so. Policies and procedures of the organizations in which one participates must also be obeyed. But compliance must be balanced with the recognition that sometimes existing laws and rules may be immoral or inappropriate and, therefore, must be challenged. Violation of a law or regulation may be ethical when that law or rule has inadequate moral basis or when it conflicts with another law judged to be more important. If one decides to violate a law or rule because it is viewed as unethical, or for any other reason, one must fully accept responsibility for one's actions and for the consequences.

2.4 Accept and provide appropriate professional review.

Quality professional work, especially in the computing profession, depends on professional reviewing and critiquing. Whenever appropriate, individual members should seek and utilize peer review as well as provide critical review of the work of others.

2.5 Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks.

Computer professionals must strive to be perceptive, thorough, and objective when evaluating, recommending, and presenting system descriptions and alternatives. Computer professionals are in a position of special trust, and therefore have a special responsibility to provide objective, credible evaluations to employers, clients, users, and the public. When providing evaluations the professional must also identify any relevant conflicts of interest, as stated in imperative 1.3.

As noted in the discussion of principle 1.2 on avoiding harm, any signs of danger from systems must be reported to those who have opportunity and/or responsibility to resolve them. See the guidelines for imperative 1.2 for more details concerning harm,including the reporting of professional violations.

2.6 Honor contracts, agreements, and assigned responsibilities.

Honoring one's commitments is a matter of integrity and honesty. For the computer professional this includes ensuring that system elements perform as intended. Also, when one contracts for work with another party, one has an obligation to keep that party properly informed about progress toward completing that work.

A computing professional has a responsibility to request a change in any assignment that he or she feels cannot be completed as defined. Only after serious consideration and with full disclosure of risks and concerns to the employer or client, should one accept the assignment. The major underlying principle here is the obligation to accept personal accountability for professional work. On some occasions other ethical principles may take greater priority.

A judgment that a specific assignment should not be performed may not be accepted. Having clearly identified one's concerns and reasons for that judgment, but failing to procure a change in that assignment, one may yet be obligated, by contract or by law, to proceed as directed. The computing professional's ethical judgment should be the final guide in deciding whether or not to proceed. Regardless of the decision, one must accept the responsibility for the consequences.

However, performing assignments "against one's own judgment" does not relieve the professional of responsibility for any negative consequences.

2.7 Improve public understanding of computing and its consequences.

Computing professionals have a responsibility to share technical knowledge with the public by encouraging understanding of computing, including the impacts of computer systems and their limitations. This imperative implies an obligation to counter any false views related to computing.

2.8 Access computing and communication resources only when authorized to do so.

Theft or destruction of tangible and electronic property is prohibited by imperative 1.2 - "Avoid harm to others." Trespassing and unauthorized use of a computer or communication system is addressed by this imperative. Trespassing includes accessing communication networks and computer systems, or accounts and/or files associated with those systems, without explicit authorization to do so. Individuals and organizations have the right to restrict access to their systems so long as they do not violate the discrimination principle (see 1.4). No one should enter or use another's computer system, software, or data files without permission. One must always have appropriate approval before using system resources, including communication ports, file space, other system peripherals, and computer time.

3. ORGANIZATIONAL LEADERSHIP IMPERATIVES.

As an ACM member and an organizational leader, I will ....

> BACKGROUND NOTE: *This section draws extensively from the draft IFIP Code of Ethics,especially its sections on organizational ethics and international concerns. The ethical obligations of organizations tend to be neglected in most codes of professional conduct, perhaps because these codes are written from the perspective of the individual member. This dilemma is addressed by stating these imperatives from the perspective of the organizational leader. In this context "leader" is viewed as any organizational member who has leadership or educational responsibilities. These imperatives generally may apply to organizations as well as their leaders. In this context "organizations" are corporations, government agencies,and other "employers," as well as volunteer professional organizations.*

3.1 Articulate social responsibilities of members of an organizational unit and encourage full acceptance of those responsibilities.

Because organizations of all kinds have impacts on the public, they must accept responsibilities to society. Organizational procedures and attitudes oriented toward quality and the welfare of society will reduce harm to members of the public, thereby serving public interest and fulfilling social responsibility. Therefore,organizational leaders must encourage full participation in meeting social responsibilities as well as quality performance.

3.2 Manage personnel and resources to design and build information systems

that enhance the quality of working life.

Organizational leaders are responsible for ensuring that computer systems enhance, not degrade, the quality of working life. When implementing a computer system, organizations must consider the personal and professional development, physical safety, and human dignity of all workers. Appropriate human-computer ergonomic standards should be considered in system design and in the workplace.

3.3 Acknowledge and support proper and authorized uses of an organization's computing and communication resources.

Because computer systems can become tools to harm as well as to benefit an organization, the leadership has the responsibility to clearly define appropriate and inappropriate uses of organizational computing resources. While the number and scope of such rules should be minimal, they should be fully enforced when established.

3.4 Ensure that users and those who will be affected by a system have their needs clearly articulated during the assessment and design of requirements; later the system must be validated to meet requirements.

Current system users, potential users and other persons whose lives may be affected by a system must have their needs assessed and incorporated in the statement of requirements. System validation should ensure compliance with those requirements.

3.5 Articulate and support policies that protect the dignity of users and others affected by a computing system.

Designing or implementing systems that deliberately or inadvertently demean individuals or groups is ethically unacceptable. Computer professionals who are in decision making positions should verify that systems are designed and implemented to protect personal privacy and enhance personal dignity.

3.6 Create opportunities for members of the organization to learn the principles and limitations of computer systems.

This complements the imperative on public understanding (2.7). Educational opportunities are essential to facilitate optimal participation of all organizational members. Opportunities must be available to all members to help them improve their knowledge and skills in computing, including courses that familiarize them with the consequences and limitations of particular types of systems.In particular, professionals must be made aware of the dangers of building systems around oversimplified models, the improbability of anticipating and designing for every possible operating condition, and other issues related to the complexity of this profession.

4. COMPLIANCE WITH THE CODE.

As an ACM member I will ....

4.1 Uphold and promote the principles of this Code.

The future of the computing profession depends on both technical and ethical excellence. Not only is it important for ACM computing professionals to adhere to the principles expressed in this Code, each member should encourage and support adherence by other members.

4.2 Treat violations of this code as inconsistent with membership in the ACM.

Adherence of professionals to a code of ethics is largely a voluntary matter. However, if a member does not follow this code by engaging in gross misconduct, membership in ACM may be terminated.

This Code and the supplemental Guidelines were developed by the Task Force for the Revision of the ACM Code of Ethics and Professional Conduct: Ronald E. Anderson, Chair, Gerald Engel, Donald Gotterbarn, Grace C. Hertlein, Alex Hoffman, Bruce Jawer, Deborah G. Johnson, Doris K. Lidtke, Joyce Currie Little, Dianne Martin, Donn B. Parker, Judith A. Perrolle, and Richard S. Rosenberg. The Task Force was organized by ACM/SIGCAS and funding was provided by the ACM SIG Discretionary Fund. This Code and the supplemental Guidelines were adopted by the ACM Council on October 16, 1992.

Appendix A

Modeling

A.1 ASM Charts

Large state diagrams get ugly, due to the need to label each input and output, even if they are irrelevant to that state. Algorithmic State Machine (ASM) Charts take care of this problem, and thus are much easier to read. Only relevant inputs and asserted signals are listed.

ASM Charts are much like flow charts, and consist of states and decisions connected by directed lines. There are three types of symbols used:

rectangle State block. The name goes on the upper left (either above or in the box). The state code goes on the upper right, and the asserted output goes on the lower inside of the box. Each box is a state, and the system remains in it through a whole clock cycle.

diamond Decision block. The condition goes inside, and the potential answers (usually T/F) goes on the lines exiting the diamond.

Rounded rectangle Conditional outputs. If a Mealy style output is used, then these conditional outputs may be asserted in the conditional output block without changing state.

Basic Rules:

- ASMs are deterministic state machines, but they can include parallel conditions, so long as no state is non-deterministic. Both serial and parallel conditions are performed simultaneously.
- Lines must point to a symbol not another line.
- At each moment, the system may only be in a state block. The downstream decision blocks and conditional blocks are calculated during the cycle, so the transition can take place at the clock pulse.

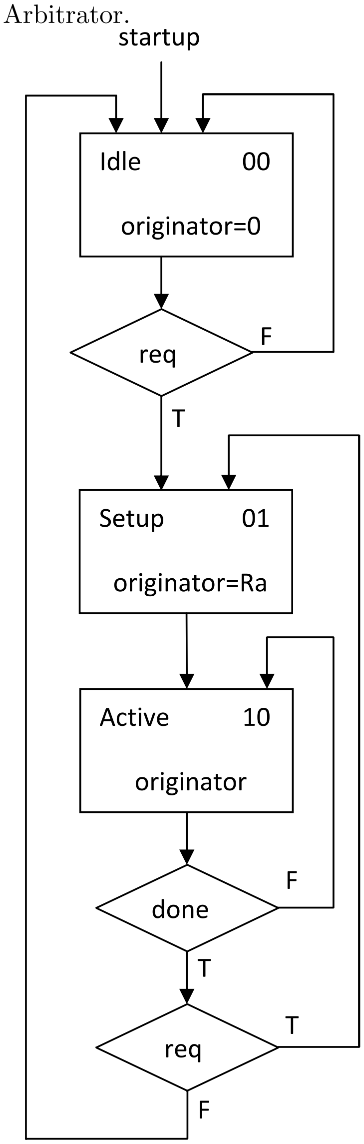
- Only state blocks can have multiple inputs.

Example: Bus Arbitrator

Imagine we are to design a bus arbitrator. The basic scheme is that there is to be a request line, *req*, that any device can assert as long as it is not currently asserted. When the request line is asserted, the requestor also puts their identifier on the requestor's address line , *Ra*. If the system is idle (no one transferring) then the system changes first to a setup state, which sets the originator to the address of the requestor. When the requestor sees they have the bus, the requestor de-asserts the request line and request address. The requestor is now the transmitter and we go to the active state. Since the request lines are free, another device can que a request. Note only one can que a request as no one else can request once the line is asserted. When the transmitter is done, the done line is asserted and the request line is checked. If there is a waiting request this system becomes the transmitter, else the system goes idle.

This is shown in the ASM chart, figure A.1. Let's consider this chart. We start in the idle state, where the originator is set to zero. This makes the assumption there is no device with address 0, which is a common assumption. The alternative is to add a busy line and assert it[1]. If no request comes then we stay in idle. If a request comes we go to setup and copy Ra to originator. The remote machine must drop the request so we do not list this. In the remote devices logic when their address gets copied to address, they would switch states and do this. In any case, the next cycle the system goes to the active state and continues to assert the originator. When the remote system signals done, the request line is checked.

Figure A.1: ASM Chart of Bus Arbitrator.



[1]This is inefficient because one extra line is another bit, which could double the number of addresses, then minus one to account for the unused code

A.2 Block Diagrams

Appendix B

Using SciLab

B.1 Basics

When you first start SciLab you will see something like

```
                  ==========
                   scilab-2.7.2
        Copyright (C) 1989-2003 INRIA/ENPC
                   ==========



Startup execution:
  loading initial environment

-->
```

The arrow "−−>" is the command prompt. SciLab, like MatLab, is a command line interface to a mathematics programming environment. To get started lets do a calculation.

```
3+(2+5*4)/11
```

SciLab performs the calculation and displays the answer.

```
 ans  =

    5.
```

Now lets define a simple variable.

```
a=2
```

SciLab responds with

```
 a  =

    2.
```

Notice anything similar? The response is almost the same but "ans" has been replaced by the variable name "a". In fact it is even more similar than that. When no assignment ("name=") is given, SciLab automatically assigns the result to the variable "ans". Try using it.

```
a*ans
```

SciLab will tell you that "ans" is now 10. Lets move on and define a matrix. Type the following

```
  A=[1,2;3,4;5,6]
```

and press enter. Commas are used to separate elements and semicolons are used to separate rows. Note that you could also have entered "A" using the alternate notation

```
  A=[[1 2];[3 4];[5 6]]
```

or even (command prompt shown so you won't think something is wrong when it automatically appears, also you do not need to space over like I do to enter the numbers, I just find it easier to read)

```
-->  A=[[1 2]
-->     [3 4]
-->     [5 6]]
```

Thus spaces work like commas and returns work like semicolons. In any case, SciLab should respond by showing you that it has created the matrix variable as follows

```
 A  =

!   1.    2. !
!   3.    4. !
!   5.    6. !
```

The variable "A" is now defined and can be used. For instance we might want to define "B" to be "A+A". Do this by typing

```
  B=A+A
```

SciLab will add the matrices and define "B" to be the result, showing you the answer.

```
 B  =

!   2.      4.  !
!   6.      8.  !
!   10.     12. !
```

This mode is useful for doing simple calculations and testing output. We will refer to it as the interactive mode. Since SciLab has an interactive mode that is command driven, it is reasonable to assume it would have a programming interface (we will refer to it as the programming mode). I will show the use of programming mode later.

B.2 Programming

Scilab is a Matlab look alike. We will use Scilab as it is free and open source. If you know one, you can figure the other out easily.

There are two basic ways to interact with Scilab: command line execution, and script files[1]. Yes there are others such as compiled programs (MEX-files in Matlab), a GUI - Scicos (Simulink in Matlab), and several interfacing programs, but they are not relevant to us at the moment. We will primarily be concerned with the use of script files, because they are the most helpful. Command line execution is really just for quick operations and checking of segments of code. Scilab syntax is a high level programming language that interacts with a series of numerical libraries (most notably LinPack[2], EisPack[3], and BLAS[4]). Like most programming languages we have two types of programs that can be written. A regular program, which is written as you would type commands on the command line, is the most basic type and is often the way you will start homework problems and other projects. I am not going to try to be exhaustive, as the help files are quite nice (type "help" for a searchable help file or "help command" for help on some command (obviously you must replace command with the actual command you want help on). Functions, which are subprograms called by another program (even by other functions), are probably the most useful, as they allow you to extend the language by defining new operations. One of the side goals of this book is for you to walk away with a basic familiarity of the tools that you can use to do a variety of tasks. So how do you specify which you want? You will get a regular program unless you start the script file with the command function. The syntax is

```
function [a,b,...,c]=name(x,y,..., z)
```

Note you can return multiple values. You also have several command structures: for, while, and if-elseif-else. To see how these work let's make up a program.

Listing B.1: Quadratic Function

```
function [y]=quadratic(x,a,b,c)
  y=a*x^2+b*x+c;
endfunction
```

[1]Scilab distinguishes between directly executed files (.sce) and function libraries (.sci) which must be called by something else. Matlab just uses M-files, a .m extension.

[2]Linear Algebra Package

[3]Eigen Systems Package, i.e. eigenvalues and eigenvectors, for basic systems, symmetric systems, generalized, and singular value decompositions.

[4]Basic Linear Algebra Subroutines, of which there are 3 levels: 1 is vector-vector operations, 2 is matrix-vector operations, and 3 is matrix-matrix operations.

Figure B.1: Scicos Sources

Figure B.2: Scicos Sinks

Now we want to run it. Create the following script file in Scipad (Scilab's editor) then save and execute.

Listing B.2: Graphing a Quadratic

```
exec("quadratic.sci");
x=-5:.01:5;
y=quadratic(x,1,0,-1);
plot(x,y,'r-')
plot([min(x);max(x)],[0;0],'k-')
plot([0;0],[min(y)-4;max(y)+1],'k-')
```

The resulting graph is

B.3 Scicos

Scilab's graphical environment for evaluating block diagrams is called Scicos. You can pull up the Scicos editor by typing

```
--> scicos
```

A new window will open with a bunch of menus. First we want to get our palettes so we can start adding things. Click on the "Palette" menu and select "Palettes". A menu will come up and you should select "sources". Pull up the palettes again and select "sinks". From the sources drag a "sinusoid generator" and a clock (needed to time the sampling of the sinusoid for the graph). From the sinks drag the graph symbol in the lower left. You can connect them by clicking on the arrows (source then sink) and a line will be dragged automatically. By double clicking you can edit the parameters of a block. We will leave the defaults for now. Select the "Simulate" menu and then click on "Setup". A window will pop up. Change the "Final integration time" to 30 or it will run forever. Now select the simulate menu again and click run. A graph will appear and plot a sine function for 30 units. If you did not change the final time it would just keep scrolling (a really annoying effect) and you would have to click "Stop" on the far right of the menu bar on the main Scicos window.

Figure B.3: Scicos Integration Model of Disease

Figure B.4: Scicos Plot of Disease Spread

Appendix C

SPICE

SPICE is an acronym (come on this is engineering of course it is an cutsie acronym to try to make it memorable) that stands for Simulation Program with Integrated Circuit Emphasis. SPICE is often used as a simulator for circuit layout, and is thus usually incorporated into EDA (Electronic Design Analysis tools) like OrCAD or Mentor Graphics. I will cover a separate SPICE tool called 5SPICE, which was designed for prototyping, and is free for educational use.

SPICE allows us to simulate a circuit without having to build it. You can select actual component families, operating temperatures, and other operating conditions. This allows a more accurate simulation, and thus less prototyping is needed. This saves a lot of time and money, while improving the quality of the design, by allowing design changes in the simulation.

We will create an inverter gate using a transistor and some resistors. First we need to put a transistor in our blank schematic. We do this by selecting the NPN transistor symbol (if you don't know it, the name will appear in the tool tip when you hover) off the diode button on the right button bar. When we have placed it by left clicking, we right click twice (once to end adding transistors and the second to edit the transistor we place). A dialog will appear, from which you should select "BC848C" then click "ok" to accept. Your screen should look like below.

We now need to add two resistors to keep the current flows within reason. Resistors are selected from the resistor button on the right button bar. Place two resistors on the sheet. Right click to end adding and then right click one to edit. Change the resistance to 100 (ohms is implicit) then click ok. Move this one to the collector, to keep the current from collector to emitter to a reasonable value. Now edit the other so it is also 100 ohms and click ok. Right click it again and rotate. Drag to the gate. You schematic should look as follows.

We know need to add several items from the power source button. Start with a DC source to supply power to the transistor. We will need to edit it so it has 5 volts. Since we don't want to drag wires all over the schematic (we could, I just think it looks ugly) we will use two voltage points to connect the DC source to the resistor on the source. Let's edit them to name it VpVCC. Now add an AC voltage source. When you edit its properties there will be several tabs so select the one called transient. Select square wave, 5 volts peak to peak, 2.5 volts bias (so it goes 0-5 not -2.5 to 2.5), a frequency of 1, and a rise time of .1. I also set the delay to 1 so it stays at 0 volts for a second before transitioning. We will again add two voltage points to keep things neat, this time putting them on the input signal and calling them VpSignal. Now add three grounds where marked on the diagram.

Now we need to add two test points, so we graph something. You can only plot from test points. To keep it neat I used wires to connect them.

We now need to set the defaults for the simulator. From the Analyze menu, select "Project Defaults..." then select the Analysis tab. Pick transient analysis using Vs1 and simulate from 0 to 10 seconds.

Now select the Graph/Table tab and under Plots, select TPv1 with Left axis and TPv2 with Left axis. Note if you don't select the axis it is off by default. Then set the vertical axis to go from -1 to 6 so there is some room around the graph (again for aesthetics). Save and Run.

You will get a graph like below after a few seconds. Note the graph shows we have an inverter. You will also note the inverted signal does not go to zero exactly due to the resistors. If you play with the resistances you will get some interesting results. Try and see. You should also see the cyan circle with the vertical line at the top of the graph. It is the measurement device, and it is the selector for the x,y data on the top right. Try moving it and reading the voltages.

Appendix D

Probability and Statistics

D.1 Quick Overview

Let X be a set of n numbers, $\{x_0, x_1, \ldots, x_{n-1}\}$.

The (algebraic) mean of X is denoted by μ_X and is given by

$$\begin{aligned} \mu_X &= E[X] && \text{(D.1)} \\ &= \frac{\sum_{i=0}^{n-1} x_i}{n} && \text{(D.2)} \end{aligned}$$

and is the average value. We sometimes also speak of the mode (the most frequently occurring value), the median (middle term of the ordered sequence if the sequence length is odd, or the average of the two middle terms of the ordered sequence if the sequence length is even), the geometric mean

$$\sqrt[n]{\prod_{i=0}^{n-1} x_i}, \tag{D.3}$$

or even the harmonic mean

$$\frac{1}{\sum_{i=0}^{n-1} \frac{1}{x_i}}. \tag{D.4}$$

The variance is given by

$$\begin{aligned} var(X) &= E[(X - E[X])^2] && \text{(D.5)} \\ &= \sigma_X^2 && \text{(D.6)} \\ &= \frac{\sum_{i=0}^{n-1} (x_i - \mu)^2}{n} && \text{(D.7)} \\ &= \frac{\sum_{i=0}^{n-1} x_i^2}{n} - \mu^2. && \text{(D.8)} \end{aligned}$$

Note the standard deviation is the square root of the variance. It the variance is being calculated you need to divide by $n-1$ (the degrees of freedom) rather than n (the size of the set).

The coefficient of variation is

$$c_v = \frac{\sigma_X}{\mu_X} \tag{D.9}$$

and is useful for doing comparisons of distributions because it is a dimensionless comparison of the variability of a set (or random variable).

The covariance of X and Y (two different sets) is

$$\begin{aligned} cov(X,Y) &= E[(X-E[X])(Y-E[Y])] && \text{(D.10)} \\ &= \sigma^2_{XY} && \text{(D.11)} \\ &= \frac{\sum_{i=0}^{n-1}(x_i-\mu_X)(y_i-\mu_Y)}{n} && \text{(D.12)} \\ &= \frac{\sum_{i=0}^{n-1} x_i y_i}{n} - \mu_X\mu_Y. && \text{(D.13)} \end{aligned}$$

The correlation (measure of linear dependence) is

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X\sigma_Y} \tag{D.14}$$

Appendix E

Compound Interest Factors

I created the following tables in Excel using the formulas in Chapter 5. They can be used for quick calculations of formulas when a computer is not available.

r= 2.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9804	1.0200	0.9804	1.0200	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.9612	1.0404	1.9416	0.5150	2.0200	0.4950	0.9612	1.0000	0.4950
3	0.9423	1.0612	2.8839	0.3468	3.0604	0.3268	2.8458	3.0200	0.9868
4	0.9238	1.0824	3.8077	0.2626	4.1216	0.2426	5.6173	6.0804	1.4752
5	0.9057	1.1041	4.7135	0.2122	5.2040	0.1922	9.2403	10.2020	1.9604
6	0.8880	1.1262	5.6014	0.1785	6.3081	0.1585	13.6801	15.4060	2.4423
7	0.8706	1.1487	6.4720	0.1545	7.4343	0.1345	18.9035	21.7142	2.9208
8	0.8535	1.1717	7.3255	0.1365	8.5830	0.1165	24.8779	29.1485	3.3961
9	0.8368	1.1951	8.1622	0.1225	9.7546	0.1025	31.5720	37.7314	3.8681
10	0.8203	1.2190	8.9826	0.1113	10.9497	0.0913	38.9551	47.4860	4.3367
15	0.7430	1.3459	12.8493	0.0778	17.2934	0.0578	85.2021	114.6708	6.6309
20	0.6730	1.4859	16.3514	0.0612	24.2974	0.0412	144.6003	214.8685	8.8433
25	0.6095	1.6406	19.5235	0.0512	32.0303	0.0312	214.2592	351.5150	10.9745
30	0.5521	1.8114	22.3965	0.0446	40.5681	0.0246	291.7164	528.4040	13.0251
40	0.4529	2.2080	27.3555	0.0366	60.4020	0.0166	461.9931	1020.0992	16.8885
50	0.3715	2.6916	31.4236	0.0318	84.5794	0.0118	642.3606	1728.9701	20.4420
60	0.3048	3.2810	34.7609	0.0288	114.0515	0.0088	823.6975	2702.5770	23.6961
70	0.2500	3.9996	37.4986	0.0267	149.9779	0.0067	999.8343	3998.8956	26.6632
80	0.2051	4.8754	39.7445	0.0252	193.7720	0.0052	1166.7868	5688.5979	29.3572
90	0.1683	5.9431	41.5869	0.0240	247.1567	0.0040	1322.1701	7857.8328	31.7929
100	0.1380	7.2446	43.0984	0.0232	312.2323	0.0032	1464.7527	10611.6153	33.9863

r= 3.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9709	1.0300	0.9709	1.0300	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.9426	1.0609	1.9135	0.5226	2.0300	0.4926	0.9426	1.0000	0.4926
3	0.9151	1.0927	2.8286	0.3535	3.0909	0.3235	2.7729	3.0300	0.9803
4	0.8885	1.1255	3.7171	0.2690	4.1836	0.2390	5.4383	6.1209	1.4631
5	0.8626	1.1593	4.5797	0.2184	5.3091	0.1884	8.8888	10.3045	1.9409
6	0.8375	1.1941	5.4172	0.1846	6.4684	0.1546	13.0762	15.6137	2.4138
7	0.8131	1.2299	6.2303	0.1605	7.6625	0.1305	17.9547	22.0821	2.8819
8	0.7894	1.2668	7.0197	0.1425	8.8923	0.1125	23.4806	29.7445	3.3450
9	0.7664	1.3048	7.7861	0.1284	10.1591	0.0984	29.6119	38.6369	3.8032
10	0.7441	1.3439	8.5302	0.1172	11.4639	0.0872	36.3088	48.7960	4.2565
15	0.6419	1.5580	11.9379	0.0838	18.5989	0.0538	77.0002	119.9638	6.4500
20	0.5537	1.8061	14.8775	0.0672	26.8704	0.0372	126.7987	229.0125	8.5229
25	0.4776	2.0938	17.4131	0.0574	36.4593	0.0274	182.4336	381.9755	10.4768
30	0.4120	2.4273	19.6004	0.0510	47.5754	0.0210	241.3613	585.8472	12.3141
40	0.3066	3.2620	23.1148	0.0433	75.4013	0.0133	361.7499	1180.0420	15.6502
50	0.2281	4.3839	25.7298	0.0389	112.7969	0.0089	477.4803	2093.2289	18.5575
60	0.1697	5.8916	27.6756	0.0361	163.0534	0.0061	583.0526	3435.1146	21.0674
70	0.1263	7.9178	29.1234	0.0343	230.5941	0.0043	676.0869	5353.1355	23.2145
80	0.0940	10.6409	30.2008	0.0331	321.3630	0.0031	756.0865	8045.4340	25.0353
90	0.0699	14.3005	31.0024	0.0323	443.3489	0.0023	823.6302	11778.2968	26.5667
100	0.0520	19.2186	31.5989	0.0316	607.2877	0.0016	879.8540	16909.5911	27.8444

r= 4.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9615	1.0400	0.9615	1.0400	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.9246	1.0816	1.8861	0.5302	2.0400	0.4902	0.9246	1.0000	0.4902
3	0.8890	1.1249	2.7751	0.3603	3.1216	0.3203	2.7025	3.0400	0.9739
4	0.8548	1.1699	3.6299	0.2755	4.2465	0.2355	5.2670	6.1616	1.4510
5	0.8219	1.2167	4.4518	0.2246	5.4163	0.1846	8.5547	10.4081	1.9216
6	0.7903	1.2653	5.2421	0.1908	6.6330	0.1508	12.5062	15.8244	2.3857
7	0.7599	1.3159	6.0021	0.1666	7.8983	0.1266	17.0657	22.4574	2.8433
8	0.7307	1.3686	6.7327	0.1485	9.2142	0.1085	22.1806	30.3557	3.2944
9	0.7026	1.4233	7.4353	0.1345	10.5828	0.0945	27.8013	39.5699	3.7391
10	0.6756	1.4802	8.1109	0.1233	12.0061	0.0833	33.8814	50.1527	4.1773
15	0.5553	1.8009	11.1184	0.0899	20.0236	0.0499	69.7355	125.5897	6.2721
20	0.4564	2.1911	13.5903	0.0736	29.7781	0.0336	111.5647	244.4520	8.2091
25	0.3751	2.6658	15.6221	0.0640	41.6459	0.0240	156.1040	416.1477	9.9925
30	0.3083	3.2434	17.2920	0.0578	56.0849	0.0178	201.0618	652.1234	11.6274
40	0.2083	4.8010	19.7928	0.0505	95.0255	0.0105	286.5303	1375.6379	14.4765
50	0.1407	7.1067	21.4822	0.0466	152.6671	0.0066	361.1638	2566.6771	16.8122
60	0.0951	10.5196	22.6235	0.0442	237.9907	0.0042	422.9966	4449.7671	18.6972
70	0.0642	15.5716	23.3945	0.0427	364.2905	0.0027	472.4789	7357.2615	20.1961
80	0.0434	23.0498	23.9154	0.0418	551.2450	0.0018	511.1161	11781.1244	21.3718
90	0.0293	34.1193	24.2673	0.0412	827.9833	0.0012	540.7369	18449.5833	22.2826
100	0.0198	50.5049	24.5050	0.0408	1237.6237	0.0008	563.1249	28440.5926	22.9800

r= 5.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9524	1.0500	0.9524	1.0500	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.9070	1.1025	1.8594	0.5378	2.0500	0.4878	0.9070	1.0000	0.4878
3	0.8638	1.1576	2.7232	0.3672	3.1525	0.3172	2.6347	3.0500	0.9675
4	0.8227	1.2155	3.5460	0.2820	4.3101	0.2320	5.1028	6.2025	1.4391
5	0.7835	1.2763	4.3295	0.2310	5.5256	0.1810	8.2369	10.5126	1.9025
6	0.7462	1.3401	5.0757	0.1970	6.8019	0.1470	11.9680	16.0383	2.3579
7	0.7107	1.4071	5.7864	0.1728	8.1420	0.1228	16.2321	22.8402	2.8052
8	0.6768	1.4775	6.4632	0.1547	9.5491	0.1047	20.9700	30.9822	3.2445
9	0.6446	1.5513	7.1078	0.1407	11.0266	0.0907	26.1268	40.5313	3.6758
10	0.6139	1.6289	7.7217	0.1295	12.5779	0.0795	31.6520	51.5579	4.0991
15	0.4810	2.0789	10.3797	0.0963	21.5786	0.0463	63.2880	131.5713	6.0973
20	0.3769	2.6533	12.4622	0.0802	33.0660	0.0302	98.4884	261.3191	7.9030
25	0.2953	3.3864	14.0939	0.0710	47.7271	0.0210	134.2275	454.5420	9.5238
30	0.2314	4.3219	15.3725	0.0651	66.4388	0.0151	168.6226	728.7770	10.9691
40	0.1420	7.0400	17.1591	0.0583	120.7998	0.0083	229.5452	1615.9955	13.3775
50	0.0872	11.4674	18.2559	0.0548	209.3480	0.0048	277.9148	3186.9599	15.2233
60	0.0535	18.6792	18.9293	0.0528	353.5837	0.0028	314.3432	5871.6744	16.6062
70	0.0329	30.4264	19.3427	0.0517	588.5285	0.0017	340.8409	10370.5702	17.6212
80	0.0202	49.5614	19.5965	0.0510	971.2288	0.0010	359.6460	17824.5764	18.3526
90	0.0124	80.7304	19.7523	0.0506	1594.6073	0.0006	372.7488	30092.1460	18.8712
100	0.0076	131.5013	19.8479	0.0504	2610.0252	0.0004	381.7492	50200.5031	19.2337

r= 6.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9434	1.0600	0.9434	1.0600	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.8900	1.1236	1.8334	0.5454	2.0600	0.4854	0.8900	1.0000	0.4854
3	0.8396	1.1910	2.6730	0.3741	3.1836	0.3141	2.5692	3.0600	0.9612
4	0.7921	1.2625	3.4651	0.2886	4.3746	0.2286	4.9455	6.2436	1.4272
5	0.7473	1.3382	4.2124	0.2374	5.6371	0.1774	7.9345	10.6182	1.8836
6	0.7050	1.4185	4.9173	0.2034	6.9753	0.1434	11.4594	16.2553	2.3304
7	0.6651	1.5036	5.5824	0.1791	8.3938	0.1191	15.4497	23.2306	2.7676
8	0.6274	1.5938	6.2098	0.1610	9.8975	0.1010	19.8416	31.6245	3.1952
9	0.5919	1.6895	6.8017	0.1470	11.4913	0.0870	24.5768	41.5219	3.6133
10	0.5584	1.7908	7.3601	0.1359	13.1808	0.0759	29.6023	53.0132	4.0220
15	0.4173	2.3966	9.7122	0.1030	23.2760	0.0430	57.5546	137.9328	5.9260
20	0.3118	3.2071	11.4699	0.0872	36.7856	0.0272	87.2304	279.7599	7.6051
25	0.2330	4.2919	12.7834	0.0782	54.8645	0.0182	115.9732	497.7419	9.0722
30	0.1741	5.7435	13.7648	0.0726	79.0582	0.0126	142.3588	817.6364	10.3422
40	0.0972	10.2857	15.0463	0.0665	154.7620	0.0065	185.9568	1912.6994	12.3590
50	0.0543	18.4202	15.7619	0.0634	290.3359	0.0034	217.4574	4005.5984	13.7964
60	0.0303	32.9877	16.1614	0.0619	533.1282	0.0019	239.0428	7885.4697	14.7909
70	0.0169	59.0759	16.3845	0.0610	967.9322	0.0010	253.3271	14965.5362	15.4613
80	0.0095	105.7960	16.5091	0.0606	1746.5999	0.0006	262.5493	27776.6649	15.9033
90	0.0053	189.4645	16.5787	0.0603	3141.0752	0.0003	268.3946	50851.2531	16.1891
100	0.0029	339.3021	16.6175	0.0602	5638.3681	0.0002	272.0471	92306.1343	16.3711

r= 7.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9346	1.0700	0.9346	1.0700	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.8734	1.1449	1.8080	0.5531	2.0700	0.4831	0.8734	1.0000	0.4831
3	0.8163	1.2250	2.6243	0.3811	3.2149	0.3111	2.5060	3.0700	0.9549
4	0.7629	1.3108	3.3872	0.2952	4.4399	0.2252	4.7947	6.2849	1.4155
5	0.7130	1.4026	4.1002	0.2439	5.7507	0.1739	7.6467	10.7248	1.8650
6	0.6663	1.5007	4.7665	0.2098	7.1533	0.1398	10.9784	16.4756	2.3032
7	0.6227	1.6058	5.3893	0.1856	8.6540	0.1156	14.7149	23.6289	2.7304
8	0.5820	1.7182	5.9713	0.1675	10.2598	0.0975	18.7889	32.2829	3.1465
9	0.5439	1.8385	6.5152	0.1535	11.9780	0.0835	23.1404	42.5427	3.5517
10	0.5083	1.9672	7.0236	0.1424	13.8164	0.0724	27.7156	54.5207	3.9461
15	0.3624	2.7590	9.1079	0.1098	25.1290	0.0398	52.4461	144.7003	5.7583
20	0.2584	3.8697	10.5940	0.0944	40.9955	0.0244	77.5091	299.9356	7.3163
25	0.1842	5.4274	11.6536	0.0858	63.2490	0.0158	100.6765	546.4148	8.6391
30	0.1314	7.6123	12.4090	0.0806	94.4608	0.0106	120.9718	920.8684	9.7487
40	0.0668	14.9745	13.3317	0.0750	199.6351	0.0050	152.2928	2280.5016	11.4233
50	0.0339	29.4570	13.8007	0.0725	406.5289	0.0025	172.9051	5093.2704	12.5287
60	0.0173	57.9464	14.0392	0.0712	813.5204	0.0012	185.7677	10764.5769	13.2321
70	0.0088	113.9894	14.1604	0.0706	1614.1342	0.0006	193.5185	22059.0596	13.6662
80	0.0045	224.2344	14.2220	0.0703	3189.0627	0.0003	198.0748	44415.1811	13.9273
90	0.0023	441.1030	14.2533	0.0702	6287.1854	0.0002	200.7042	88531.2204	14.0812
100	0.0012	867.7163	14.2693	0.0701	12381.6618	0.0001	202.2001	175452.3113	14.1703

r= 8.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9259	1.0800	0.9259	1.0800	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.8573	1.1664	1.7833	0.5608	2.0800	0.4808	0.8573	1.0000	0.4808
3	0.7938	1.2597	2.5771	0.3880	3.2464	0.3080	2.4450	3.0800	0.9487
4	0.7350	1.3605	3.3121	0.3019	4.5061	0.2219	4.6501	6.3264	1.4040
5	0.6806	1.4693	3.9927	0.2505	5.8666	0.1705	7.3724	10.8325	1.8465
6	0.6302	1.5869	4.6229	0.2163	7.3359	0.1363	10.5233	16.6991	2.2763
7	0.5835	1.7138	5.2064	0.1921	8.9228	0.1121	14.0242	24.0350	2.6937
8	0.5403	1.8509	5.7466	0.1740	10.6366	0.0940	17.8061	32.9578	3.0985
9	0.5002	1.9990	6.2469	0.1601	12.4876	0.0801	21.8081	43.5945	3.4910
10	0.4632	2.1589	6.7101	0.1490	14.4866	0.0690	25.9768	56.0820	3.8713
15	0.3152	3.1722	8.5595	0.1168	27.1521	0.0368	47.8857	151.9014	5.5945
20	0.2145	4.6610	9.8181	0.1019	45.7620	0.0219	69.0898	322.0246	7.0369
25	0.1460	6.8485	10.6748	0.0937	73.1059	0.0137	87.8041	601.3242	8.2254
30	0.0994	10.0627	11.2578	0.0888	113.2832	0.0088	103.4558	1041.0401	9.1897
40	0.0460	21.7245	11.9246	0.0839	259.0565	0.0039	126.0422	2738.2065	10.5699
50	0.0213	46.9016	12.2335	0.0817	573.7702	0.0017	139.5928	6547.1270	11.4107
60	0.0099	101.2571	12.3766	0.0808	1253.2133	0.0008	147.3000	14915.1662	11.9015
70	0.0046	218.6064	12.4428	0.0804	2720.0801	0.0004	151.5326	33126.0009	12.1783
80	0.0021	471.9548	12.4735	0.0802	5886.9354	0.0002	153.8001	72586.6929	12.3301
90	0.0010	1018.9151	12.4877	0.0801	12723.9386	0.0001	154.9925	157924.2327	12.4116
100	0.0005	2199.7613	12.4943	0.0800	27484.5157	0.0000	155.6107	342306.4463	12.4545

r= 9.00%

k	P/F	F/P	P/A	A/P	F/A	A/F	P/G	F/G	A/G
1	0.9174	1.0900	0.9174	1.0900	1.0000	1.0000	0.0000	0.0000	0.0000
2	0.8417	1.1881	1.7591	0.5685	2.0900	0.4785	0.8417	1.0000	0.4785
3	0.7722	1.2950	2.5313	0.3951	3.2781	0.3051	2.3860	3.0900	0.9426
4	0.7084	1.4116	3.2397	0.3087	4.5731	0.2187	4.5113	6.3681	1.3925
5	0.6499	1.5386	3.8897	0.2571	5.9847	0.1671	7.1110	10.9412	1.8282
6	0.5963	1.6771	4.4859	0.2229	7.5233	0.1329	10.0924	16.9259	2.2498
7	0.5470	1.8280	5.0330	0.1987	9.2004	0.1087	13.3746	24.4493	2.6574
8	0.5019	1.9926	5.5348	0.1807	11.0285	0.0907	16.8877	33.6497	3.0512
9	0.4604	2.1719	5.9952	0.1668	13.0210	0.0768	20.5711	44.6782	3.4312
10	0.4224	2.3674	6.4177	0.1558	15.1929	0.0658	24.3728	57.6992	3.7978
15	0.2745	3.6425	8.0607	0.1241	29.3609	0.0341	43.8069	159.5657	5.4346
20	0.1784	5.6044	9.1285	0.1095	51.1601	0.0195	61.7770	346.2236	6.7674
25	0.1160	8.6231	9.8226	0.1018	84.7009	0.0118	76.9265	663.3433	7.8316
30	0.0754	13.2677	10.2737	0.0973	136.3075	0.0073	89.0280	1181.1949	8.6657
40	0.0318	31.4094	10.7574	0.0930	337.8824	0.0030	105.3762	3309.8049	9.7957
50	0.0134	74.3575	10.9617	0.0912	815.0836	0.0012	114.3251	8500.9284	10.4295
60	0.0057	176.0313	11.0480	0.0905	1944.7921	0.0005	118.9683	20942.1348	10.7683
70	0.0024	416.7301	11.0844	0.0902	4619.2232	0.0002	121.2942	50546.9242	10.9427
80	0.0010	986.5517	11.0998	0.0901	10950.5741	0.0001	122.4306	120784.1566	11.0299
90	0.0004	2335.5266	11.1064	0.0900	25939.1842	0.0000	122.9758	287213.1583	11.0726
100	0.0002	5529.0408	11.1091	0.0900	61422.6755	0.0000	123.2335	681363.0607	11.0930

Bibliography

[1] American National Standards Institute/Electronic Industry Alliance. *Configuration Management*. ANSI/EIA 649A.

[2] American National Standards Institute/Electronic Industry Alliance. *Processes for Engineering a System*. ANSI/EIA 632.

[3] Carnegie Mellon University. *Capability Maturity Model-Integration for Software Engineering, Systems Engineering, Integrated Product and Process Development and Supplier Sourcing*. CMMI SWE/SE/IPPD/SS.

[4] Department of Defense. *Defense Acquisition Guide*. See Chapter 4.

[5] Department of Defense. *Military Handbook, Configuration Management Guidance*. MIL-HDBK-61A.

[6] Department of Defense. *Defense System Software Development*, 1985. DOD-STD-2167.

[7] Electronic Industry Alliance. *Configuration Management Data Exchange and Interoperability*. EIA 836.

[8] Electronic Industry Alliance. *Systems Engineering Capability Model*. EIA-731.

[9] Institute of Electrical and Electronic Engineers. *Application and Management of the Systems Engineering Process*. IEEE 1220.

[10] International Organization for Standardization. *Quality Management - Guidelines for Configuration Management*. ISO 10007.

[11] International Organization for Standardization/International Electrotechnical Commission. *Systems Engineering-System Life Cycle Processes*. ISO/IEC 15288.