Diogenes, a Search for

Unintended Consequences Process

A Class Project for SIE-454/554 Fall 2010

A. Terry Bahill

<http://www.sie.arizona.edu/sysengr/sie554/>

December 14, 2010

Copyright ©, 2010, Bahill

Document 5: The Concept Exploration Document

The system design process is a use-case-based iterative process. Use cases describe the required functionality of the system: the functional and nonfunctional requirements are developed while writing the use cases. These customer requirements are used to derive technical requirements and create test procedures. Some of the highest priority requirements will be made into evaluation criteria for use in tradeoff studies. The system functions (described in the use cases) are mapped to classes and blocks, and these are interconnected to reveal the system architecture. Testing is more effective if it is planned right along with the system. Basing the tests on the use cases allows development of the testing procedures to begin right at the beginning of the system design. This should help produce more complete test plans and develop customer support for the test plans. The test plans should also include system experiments that will test the logic of the state machine diagrams. The system design process is not serial; it is very iterative and there are many opportunities for parallel processing.

Using Bahill’s system design process to design a process is certainly different. Not surprisingly, some of the documents are unusual. This is the case with document 5. In this document will consider the following four decisions.

1. Alternative implementations: paper and pencil, a computer, an Intranet with all data staying behind a firewall, an Internet system or a two-week on-site short course. The choice will be based on the type of system documentation that is available and the required level of confidentiality.

2. Alternative sources of knowledge: a systems engineer, a domain expert, a Delphi or a mixed team of domain experts and systems engineers.

3. Alternative incipient processes: a process for implementing tradeoff studies, a process for implementing risk analyses, failure modes and effects analysis (FMEA), a process for implementing sensitivity analyses, a process for creating comprehensive test plans, a process for formal inspections, a process for discovering requirements, mind mapping, concept mapping and finally the SIMILAR process. If the implementation were to be an electronic circuit, then we would also consider sneak circuit analysis.

4. Alternative names: Search for unintended consequences process (SUCP), BogSatBud, Diogenes, Dropn (defects, risks, opportunities for BIST, positive and negative unintended consequences)

Potential evaluation criteria

We will need evaluation criteria for each tradeoff study that we do. Some of the criteria may be reusable in several tradeoff studies. Here are the criteria for choosing a recommended process.

**Performance**

Name of criterion: *Ease of Use.* The system should be intuitive to use. There should not be a long learning curve. The system should hide the mathematics from the user. Low complexity is desired. It should back up its data frequently. It should be deterministic, not stochastic.

Name of criterion: *Looks Forward.* The system must look forward in time. Tools can be modified to look forward or backward in time. But for tools that have been in use for decades, there is so much literature and knowledge available, that switching would be confusing.

Name of criterion: *Has Tools.* The system should have tools to help the systems engineer to discover UiCs. Brainstorming is an example of a primitive tool. A fishbone diagram is a sophisticated tool.

Name of criterion: *Inside or Outside***.** The system must look for effects of the system being designed (SystemZ) on other systems, external to SystemZ.

**Cost**

Name of criterion: *Total Life Cycle Cost* (thousands of U. S. dollars). The total lifecycle cost shall be computed from the design costs, purchase costs, installation costs, operation and maintenance costs, and retirement and replacement costs over a predetermined period of time, such as five years. As of September 8, 2010, the numbers in the tradeoff study do not include operation, maintenance, retirement and replacement costs.

Name of criterion: *Training Time* (hours per person)**.** The average amount of time that it takes to train one systems engineer or member of an inspection team to use the system.

**Company Policy**

Name of criterion: *Built-in self-test*: It is difficult to get data for this criterion because either the manufacturers do not have BiST or they do not advertize it. BiST could be passive, where the BiST system monitors outputs and check points and displays status: or BiST could be active, where the BiST system generates signals and applies them to the system inputs.

Name of criterion: *Reusability*: It is company policy that non-developmental products be considered the primary solution to addressing customer needs with customized implementations being secondary solutions. To enhance the potential reuse in other present and future BICS systems, a Reuse Design Review with participation of top management shall be scheduled before the Systems Requirements Review. During the design phase, all engineers must be alert for reuse potential. In the tradeoff study, each alternative will be given points for use of COTS products and will be deducted points for use of custom-made products. Each alternative will be given points if the analysis and results are likely to be reused in different political environments. For example, it is federal, state and local government subsidies that make renewable-energy electric-generation systems economically feasible. So the economic analyses will be much different in different countries.

Name of criterion: *Vendor Evaluation*

Description: Each vendor will be evaluated on History in the marketplace, Financial stability, Support, Responsiveness, Upgrades, Enhancements, Custom enhancements, Third party support, Technology, Product maturity, Product stability, Vendor stability, Licensing cost, Site licenses, Licensing approach, Reputation, Maintenance approach, Complementarity and whether it is a Small business.

Weight of importance (priority): 8

Basic measure: A number between 0 and 1 that indicates the “goodness” of the vendor. It also warns of specific risks.

Units: none

Measurement method: The Analyst fills out the BICS Vendor Evaluation spreadsheet.

Input: The Analyst assigns cardinal numbers (probably integers) between 1 and 10.

Scoring function: no scoring function is necessary

Output: 0 to 1

Traces to Company Policy

|  |
| --- |
| Table 5-1. Prioritization of Evaluation Criteria |
| Evaluation Criteria | Weight of Importance | Trace to |
| Ease of Use. | 8 | CuR0-1, CuR0-3 |
| Looks Forward | 10 |  |
| Has Tools | 4 | FR2-3, FR2-5 |
| Inside or Outside | 8 |  |
| Total Life Cycle Cost | 4 | CoR2 |
| Operations Cost | 5 |  |
| BiST | 10 | FR2-1 |
| Reusability | 6 | Company policy |
| Vendor Evaluation | 7 | Company policy |

Alternative architectures

In the beginning, we will investigate the following eight alternative processes: (1) the SIMILAR process, (2) a risk analyses processes, (3) a cause and effect process, (4) a tradeoff study process, (5) a comprehensive testing process, (6) a sensitivity analysis process, (7) a formal inspection process and (8) a requirements discovery process.. Later we will perform tradeoff studies for different decisions.

Alternative 1, Do Nothing, the SIMILAR Process

The SIMILAR process will play the role of the Do Nothing or status quo alternative. In its simplest implementation, it is merely educated common sense. It involves stating the problem, investigating alternatives, modeling the system, launching the system, assessing performance and re-evaluating the process [Bahill and Gissing. 1998].

***S****tate the problem.* Stating the problem properly is the most important task. The problem statement describes the customer’s needs, states the goals of the project, prescribes the system’s capabilities, delineates the scope of the system, reports the concept of operations, describes the stakeholders, lists the deliverables and presents the key decisions that must be made

***I****nvestigate alternatives.* Alternatives solutions are identified, analyzed and compared using evaluation criteria.

***M****odel the system.* Models will be developed for most alternative designs. The model for the preferred alternative will be expanded and used to help manage the system throughout its entire life cycle. Models for dynamic system will be state-based.

***I****ntegrate*. Integration means designing interfaces and bringing system elements together so they work as a whole. This requires extensive communication and coordination.

***L****aunch the system.* Launching the system means running the system and producing outputs -- making the system do what it was intended to do. The system should be designed so that the processes are iterative and many tasks are done in parallel.

***A****ssess performance.* Performance is assessed using evaluation criteria, technical performance measures and metrics -- measurement is the key. If you cannot measure it, you cannot control it. If you cannot control it, you cannot improve it.

***R****e-evaluate.* Engineers use feedback to help control systems and improve performance: it is one of the most fundamental engineering tools. Re-evaluation should be a continual and iterative process with many parallel feedback loops.

This process can be summarized with the acronym SIMILAR [Bahill and Gissing, 1998].



Figure 5-1. The SIMILAR process

The outline of this section is that we explain the SIMILAR process and then processes for risk analyses, cause and effect analysis, creating tradeoff studies, writing comprehensive testing plans and doing sensitivity analyses. These processes provide alternative concepts for a process to search for unintended, but foreseeable, consequences.

Alternative 2, Risk Analyses

Risk is an expression of the potential harm or loss associated with an activity executed in an uncertain environment [Bahill and Smith, 2009]. Since 1662, it has been written that risk had at least two components. “Fear of some harm ought to be proportional not only to the magnitude of the harm, but also to the probability of the event” [Arnauld and Nicole, 1996, p. 274]. This is the first use of the **phrases** *magnitude of harm* and *probability of the event*. There are some ancient Greek, Chinese and biblical sources that have the concept of risk; but they do not have these words. It is unlikely that any older source has the words, because probability was not invented until the seventeenth century [Pascal, 1654].

The main tools of a risk analysis are a risk register, risk plots, mitigation plan (risk waterfall chart) and a hierarchy of models.

The first step in a risk analysis is to identify things that could go wrong, the failure modes. Haimes [2009] recommends having several models one for each aspect of the system. Some models will overlap: some will not. He calls these a hierarchical holographic model (HHM). Typical aspects include business models (with purchase orders, invoices, costs, schedules, and return on investment); architectural models; use case models; behavioral/functional models; requirements models; physical structure/component models; and performance/parametric models.

Next, the effects of those failures should be explained. These things that could go wrong are the project risks. These risks could affect cost, schedule, performance, operations, safety, etc. Next the likelihood of each risk occurring should be estimated. If the project has a lot of statistical data, then the probability of each event might be calculated. But typically such data are not available, so the frequency of occurrence in some given time interval is estimated.

The risk register is a spreadsheet summary of risks: it is examined monthly (or maybe more often) by management and the customer. It lists the most important risks along with event names, consequences, likelihood, severity, risk magnitude and who is responsible. It is based on the risk database, which contains sentences, paragraphs and test results that document each item.

The estimated frequency of occurrence is Arnauld’s probability of the event: the other half of his definition of risk is the magnitude of harm, or the severity of consequences. For insurance companies that is easy: they use dollar values. Often this will not work and the severity of consequences must be estimated qualitatively. Bahill and Smith [2009] provide a process. Usually risk is quantified as the product of frequency of occurrence and severity of consequences. It is very important that the range of frequency of occurrence and the range of severity of consequences be the same.

Many other attributes, such as difficulty of detection and vulnerability, have been included in the risk definition. If a problem is difficult to detect (in testing, verification, etc.) we should worry more about it and increase the risk number [Bahill and Karnavas, 2000]. If we know that we are vulnerable in a certain area, we should worry more about that area and increase the risk number. Haimes [2009] includes the attributes probability, severity of consequences, safety, efficiency, reliability and resilience.

The search for UiCs process will probably be related to resilience analysis. Within the US government, both the Department of Homeland Security and the White House have an Office of Resilience. The premise of these offices is that no country has the resources to protect all its infrastructure systems (power, water, transportation, etc.) against unpredicted disruptions, such as terrorist attacks or natural disasters. The theory is that systems can be designed to recover from such disruptions. To accomplish this recovery the system needs four attributes: capacity, flexibility, tolerance and cohesion as described by Jackson [2010]. Capacity is the degree to which the system has been designed to absorb the disruption. Flexibility is the ability of the system to reorganize in the face of a disruption. Tolerance is the capability to degrade gradually following a disruption. Cohesion is the ability of the elements of the system to communicate, cooperate and collaborate with each other. Jackson [2010] provides about 40 heuristics that enable the designer to achieve these attributes.

The risk analysis process is highly iterative and many tasks are done in parallel. After completing a risk analysis, you should look at (1) the high-risk events, (2) the high consequence events (no matter how unlikely) and (3) estimates that have large uncertainty. In the next iteration you should focus resources on these three items [Haimes, 2009].

|  |  |
| --- | --- |
| SIMILAR process | Risk Analysis |
| State the problem | Problem statement |
| Investigate alternatives | Identify multiple possible risks |
| Model the system | Identify frequency of occurrence and severity of consequences for each risk |
| Integrate | Organize the data into one file so the numbers can be compared. |
| Launch the system | Rank order the risks and discuss this with the decision maker and domain experts |
| Assess performance | Every month the risk register is presented to management and progress in addressing each risk is shown |
| Re-evaluate | Risk analysis is highly iterative |

Alternative 3, Cause and Effect Analysis

The purpose of a cause and effect analysis is to find the root cause of a failure. The main tools of a cause and effect analysis are Ishikawa fishbone diagrams and cause and effect diagrams..

|  |  |
| --- | --- |
| SIMILAR process | Cause and Effect Analysis |
| State the problem | Write a problem statement describing symptoms of the failure (the effects). Put its label in a box on the right (the fish head) and draw a horizontal arrow (the backbone) running to it. |
| Investigate alternatives | Draw four to six main bones. Labeled them with names of the primary probable causes of the problem. |
| Model the system | Fishbone diagram |
| Integrate | Fishbone diagram |
| Launch the system | Add bones to the diagram. Draw arrows to show cause and effect relationships. Ask, “Why did this happen?” Write subcauses branching off the causes. Continue to ask, “Why?” and generate deeper levels of causes. Prioritize and identify the most important causes. |
| Assess performance | Gather data to test the most important causes. |
| Re-evaluate | Start another iteration. |

A cause makes something else happen. To find a cause we ask, "Why did that happen?" An effect happens because of a cause. To determine an effect we ask, "What happened?" Temporally, the cause comes before the effect.

Traditionally, the purpose of a cause and effect analysis (a. k. a. causal analysis, root cause analysis and Ishikawa fishbone diagrams), is to find the root cause of an event. The event (usually a failure) has already occurred and we are trying to find its cause. To determine a cause or to move backward in time, we ask *why*? “Why did this happen?” Whereas, a failure modes and effects analysis (as commonly used in risk analysis) tries to predict the future. To determine an effect or to move forward in time, we ask *what*? “If a certain failure mode were to manifest itself, what would be the effects?” “What if this happened?” “What problem could that cause?” How is similar to What. “How could that harm us?”

The following tools were designed for cause and effect analysis, to help find the root cause of a problem. Hence, they are written in the past or present tense. But maybe they could be rewritten in the future tense to help find UiCs.

A very simple cause and effect analysis merely asks, Why? Why? Why? Why? Why? For example, Why did the space shuttle Challenger blow up? Because a hot flame (6000 ºF) from the side of its right solid fuel rocket booster melted its structure. Why did the flame spring from the side of solid fuel rocket booster? Because the O-rings failed to seal in hot gasses from the rocket motor. Why did the O-rings fail to seal in the hot gasses? Because the O-rings were brittle. Why were the O-rings brittle? Because the temperature was too cold. Why did management launch the shuttle if it was too cold?

To show the difference between a cause and effect analysis looking back in time and a forward-looking search for UiCs, let us consider the Ariane missiles. The Ariane 4 missile was successful in launching satellites. However, the French thought that they could make more money if they made this missile bigger. So they built the Ariane 5. It blew up on its first launch destroying a billion dollars worth of satellites [Kunzig, 1997; Bahill and Henderson, 2005]. The French engineers did not search for UiCs, but if they had, it might have gone like this. The Ariane 5 is larger than the Ariane 4, but it uses the same software. So, what harm could that do? The software could get into an untested state. How could that happen? Being bigger, the missile will accelerate faster. So, what is the untested state? It will gain altitude quicker and will reach the attitude where it tilts from vertical to flight that is more horizontal earlier. How would that affect us? In this new state, there would be much more horizontal acceleration than in the Ariane 4. What problem could that cause? If the horizontal acceleration were still being integrated into horizontal velocity, then the 32-bit horizontal-velocity storage register would overflow. What harm would that do? If the horizontal velocity were still being used, then the CPU would encounter a fault. What would that do? That CPU might turn itself off, even though its backup CPU was already turned off. What would happen then? It would fly out of control and the range officer would have to blow it up.

System failures and UiCs are often caused by failure to identify or test for inappropriate interactions between systems [Bahill and Henderson, 2005] or failure to test functionality in certain states [Botta, Bahill and Bahill, 2006]. Part of the reason for this is the amount of work that would be entailed.

A cause and effect analysis can be put into a tree representation, as shown in Figure 1 for the sinking of the RMS Titanic [*Titanic,* 1997; Hooper, Foecke, Graham and Weihs, 2003; Bahill and Henderson, 2005].



**Figure 1.** Cause and effect analysis for the sinking of the Titanic

A more formal root cause analysis method is Juran’s [1989] method described here.

1. Analyze Symptoms: A symptom is the outward observable evidence of the effect of a failure. By analyzing these symptoms, we can understand the nature and extent of the problem to be solved.

2. Formulate Theories: A search for the root cause of failure should speculate about possible causes of the problem but never jump to a solution.

3. Test Theories: Before accepting any theory as true, a root cause analysis should systematically test it with data.

4. Identify Root Causes: A root cause is the source of the problem, which, when removed, will sharply reduce or eliminate the deficiency

For the RMS Titanic, the main symptom was that the ship sank rapidly. This then is the effect that we want to find a root cause for. Some theories that had been investigated include (1) low strength of steel in the hull plates and rivets due to poor quality, cold water (-2 ºC) or under design, (2) the ship was in the wrong place at the wrong time due to navigation decisions, poor visibility, high speed and poor look-out training, (3) the ship broke in half due to poor design of such an unprecedentedly large ship. Data would have to be collected for all alternative theories. The discovery of the Titanic wreckage at a depth of 3700 meters in 1985 allowed several theories to be tested. Simulations using finite element analysis showed that flooding of the bow section caused stresses above the material yield stress around the second expansion joint. This caused the Titanic to break in half. Metrological examination and testing showed that the hull plates were of the best steel available at the time. However, the quality of the wrought iron rivets, particularly those in the starboard bow section were of very poor quality [McCarty and Foecke*,* 2008].

Ishikawa diagrams

Ishikawa cause and effect diagrams were proposed in the 1960s by Kaoru Ishikawa, who pioneered quality management processes in the Kawasaki shipyards. His first step was to identify possible causes of failure, which were traditionally grouped into these major categories: manpower, methods, machines, materials, measurements and the environment.

Ishikawa diagrams are called fishbone diagrams, because they are drawn to look like the skeleton of a fish [Ishikawa, 1990] (see Figure 10). The head of the fish contains the problem statement. The main bones are the primary probable causes of the problem. Each of these bones can have smaller bones attached, representing subcauses. To create a fishbone diagram do the following.

1. Write a problem statement (effect). Put its label in a box on the right and draw a horizontal arrow (the backbone) running to it.

2. Draw four to six main bones. These are labeled with the names of the major categories of probable causes of failure.

3. Use brainstorming to add bones to the diagram. Draw arrows to show cause and effect relationships. About each cause ask, “Why did this happen?” Write subcauses branching off the causes. Continue to ask “Why?” and generate deeper levels of causes.

4. Prioritize [Botta and Bahill, 2007] and identify the most important causes.

5. Take corrective action.

6. Verify through testing.

We will now create an Ishikawa fishbone diagram for the example of incorporating photovoltaic solar panels into an existing commercial electric power grid. Suppose that the event of interest is that Tucson Electric Power (TEP) might fail to meet its load demand. For this event, the major categories of possible causes are performance, environment, project management, economics and government actions [Chaves and Bahill, 2011]. We will now break these down into subcauses and subsubcauses. In the following table, we will do this in both directions (cause and effect), but in the fishbone diagram we will only show the causes direction.

|  |  |
| --- | --- |
| **Performance Causes** | **Causes and Effects** |
| Solar panel output power changes 60 MW in 15 minutes  | *Cause:* Clouds cover a large area of solar panels, the clouds were produced by monsoon thunder storms*Effect:* Not enough power is produced. Grid voltage could drop, frequency could drop and breakers could trip. TEP might have to initiate a controlled brownout with load shedding. Customers could lose trust in TEP. To ameliorate these potential problems TEP must buy and operate backup generators and negotiate purchase agreements with other electric suppliers. |
| Out of phase power peaks | *Cause:* solar panel output peaks at noon, but customer load demand peaks at 5 PM.*Effect:* This creates a need for extra generating or storage capability. |
| Short to ground on the distribution grid  | *Cause:* snakes, raptors, trees, wind, automobile crashes*Effect:* damage to transformers and capacitor banks |
| Lightning strikes BIMS  | *Effect:* components are damaged, system could crash |
| Individual solar panel output decreases | *Cause:* Solar panels accumulate layers of dust*Effect:* Power production would decrease.  |
| Failure of the main power generators | *Effect:* grid could crash |
| Failure of backup generators | *Cause:* backup motor-generator (MG) sets could take longer than 10 minutes to start up, software failure, MG sets connect to the grid at the wrong phase or frequency, or violent storms could damage the MG sets*Effect:* brownout and load shedding |

|  |
| --- |
| **Environment Causes**  |
| Interference with native Indian religious ceremonies |
| Modification of animal migration paths  |
| Destruction of natural habitats or riparian areas |
| Higher than expected disposal or recycling cost  |
| Infrastructure damage from violent storms or flooding |
| **Project Management Causes** |
| Accidents causing injury to humans requiring medical attention |
| Drastic human mistakes resulting in human fatalities |
| Project costs become higher than projected  |
| Maintenance costs become higher than expected  |
| **Economic Causes** |
| Interest rates change |
| Cost of solar panels changes |
| **Government Action Causes** |
| Carbon emission regulations change  |
| Early elimination of rebates  |



Figure 4-4. Ishikawa fishbone diagram for incorporating photovoltaic solar panels into an existing commercial electric power grid. Obviously, each item in this diagram will be described with sentences and paragraphs elsewhere in the documentation.

Alternative 4, Tradeoff Studies

Humans often make poor decisions. To help them be better decision-makers, we teach systems engineers to create tradeoff studies. Tradeoff studies are broadly recognized as *the* method for simultaneously considering multiple alternatives with many criteria, and as such are recommended and mandated in the CMMI® Decision Analysis and Resolution process [CMMI, 2006]. Tradeoff studies, which are often called trade studies, involve a simultaneous mathematical consideration of many evaluation criteria for many alternatives, in parallel. An early description of a tradeoff study that summed weighted scores is given in a 1772 letter from Benjamin Franklin to Joseph Priestley [Bahill, 2007].

As with all systems engineering processes, creating a tradeoff study is a highly iterative process. Many iterations are done in each phase of the system design and the results are updated throughout [Daniels, Werner and Bahill 2001; Smith, Son, Piattelli-Palmarini and Bahill 2007]. In early iterations, this exploration helps produce the incipient system architecture [Rechtin 2000]. The tradeoff study is rewritten continually as more information becomes available.

The fundamental task of a tradeoff study is quantifying human values and preferences. The seminal academic descriptions of tradeoff studies appeared in Keeney and Raiffa [1976] and Edwards [1977]. Edwards focused on the numerical determination of subjective values. Keeney and Raiffa are best known for their axiomatic derivation of value and utility functions from conditions of preferential or utility independence. Tradeoff studies are prescribed in industry for choosing and ranking alternative concepts, designs, processes, hardware and techniques.



Figure 5-x. The tradeoff study process.

The components of a tradeoff study, include the (1) problem statement, (2) evaluation criteria, (3) weights of importance, (4) alternate solutions, (5) evaluation data, (6) scoring functions, (7) normalized scores, (8) combining functions, (9) preferred alternatives and (10) sensitivity analysis [Daniels, Werner and Bahill 2001; Smith, Son, Piattelli-Palmarini and Bahill 2007]. The tradeoff study process is highly iterative and many tasks are done in parallel.

|  |  |
| --- | --- |
| SIMILAR process | Tradeoff Study |
| State the problem | Problem statement |
| Investigate alternatives | Evaluation criteria, weights of importance, alternate solutions, evaluation data, scoring functions, scores, combining functions, preferred alternatives |
| Model the system | Each of the alternatives may have to be simulated in order to obtain reliable evaluation data. |
| Integrate | Combine all data into one representation and highlight the preferred alternatives |
| Launch the system | Create a spreadsheet or a database that contains all of the comparison data. Then compute numbers. |
| Assess performance | Present the results to the decision maker and see if he or she concurs. Perform a sensitivity analysis of the tradeoff study. |
| Re-evaluate | The tradeoff study process is not a serial process, many tasks are done in parallel and it is highly iterative. |

Alternative 5, Creating a Test Plan

The system design process is a use-case-based iterative process [Bahill, 2010]. It starts with a problem statement. Next, we make a rough schedule of who does what and when. Then we write the use cases that describe the required behavior of the system. While we are writing the use cases, we develop functional and nonfunctional requirements. These requirements must be verified. Design of tests can start as soon as the use cases are written. The systems engineers then derive the technical requirements and the test engineers create the test requirements. Now that we have some requirements, we can use them to form evaluation criteria that will be used in the tradeoff studies. Designing the system test plan is very iterative: it is not a serial process and as many tests as possible should be scheduled to be done in parallel. It should be based on the use cases.

|  |  |
| --- | --- |
| SIMILAR process | Test Plan |
| State the problem | Use cases |
| Investigate alternatives | Find all of the requirements that must be tested |
| Model the system | Find evaluation criteria and measures of effectiveness |
| Integrate | Roll up all of the subtests into tests |
| Launch the system | Perform the tests, doing as many tests in parallel as possible |
| Assess performance | Compile results of all the tests and discuss with the decision makers |
| Re-evaluate | Regression testing is iterative and cumulative |

Alternative 6, Sensitivity Analyses

A sensitivity analysis is a powerful technique for understanding systems. A sensitivity analysis identifies the most important parameters in a tradeoff study, often these are the cost drivers that are worthy of further investment [Smith, Szidarovszky, Karnavas and Bahill, 2008]. The earliest sensitivity analyses that we have found are the genetics studies on the pea reported by Gregor Mendel in 1865 [Stern and Sherwoods, 1966] and the statistics studies on the Irish hops crops by Gosset writing under the pseudonym Student around 1890 [Box, 1981].

In a sensitivity analysis, one or many parameters are mathematically or experimentally changed and then changes in outputs or performance indices are observed.

A sensitivity analysis should be performed anytime a model is created, a set of requirements is written, a system is designed, a decision is made, a tradeoff study iteration is completed, a risk analysis is originated or when you want to discover cost drivers. In a sensitivity analysis, values of parameters or inputs (or architectural features) are changed and changes in outputs or performance indices are measured. The results of a sensitivity analysis can be used to validate a model, warn of unrealistic model behavior, point out important assumptions, help formulate model structure, simplify a model, suggest new experiments, guide future data collection efforts, suggest accuracy for calculating parameters, adjust numerical values of parameters, choose an operating point, allocate resources, detect critical criteria, suggest tolerance for manufacturing parts and identify cost drivers.

A sensitivity analysis tells which parameters are the most important and most likely to affect system behavior and/or predictions of the model. Following a sensitivity analysis, values of critical parameters can be refined while parameters that have little effect can be simplified or ignored. In the manufacturing environment, they can be used to allocate resources to critical parts allowing casual treatment of less sensitive parts. If the sensitivity coefficients are calculated as functions of time, it can be seen *when* each parameter has the greatest effect on the output function of interest. This can be used to adjust numerical values for the parameters. The values of the parameters should be chosen to match the physical data at the times when they have the most effect on the output.

|  |  |
| --- | --- |
| SIMILAR process | Sensitivity Analysis |
| State the problem | Problem statement |
| Investigate alternatives | In a sensitivity analysis, selected system parameters are mathematically or experimentally changed and then changes in outputs or performance indices are computed. |
| Model the system | A sensitivity analysis can be done on a computer simulation or a physical system. |
| Integrate | Find the cost drivers |
| Launch the system | Compute the sensitivities for all of the parameters and discuss this with the decision makers. |
| Assess performance | Show the most important parameters to the decision maker and see if there are any surprises. |
| Re-evaluate | The sensitivity analysis process is very iterative and many tasks are done in parallel. |

Alternative 7, Formal Inspection Process

Formal software inspections were created by Michael Fagan in the 1970s. He emphasized that the cost of fixing a defect rises dramatically the later it is found in the development lifecycle of the program. Experience has shown that it costs about one man-hour to fix defects detected during inspection. It costs about three man-hours to fix defects detected during testing. It costs thirty man-hours to fix defects detected after release of the product. [Requirements engineers use the term inspection in a different sense. They say that each requirement must be verified by logical argument, inspection, modeling, simulation, analysis, test or demonstration. To them inspection means merely observing the object.]

For a system design, the material to be inspected will be use cases, tradeoff studies, verification and validation documents including test plans, and system designs for example sequence diagrams, class diagrams, state machine diagrams, block definition diagrams, parametric diagrams, etc. In this document, these will be called work products. The following is a more detailed listing of the work products broken down by the RUP’s like cycle phases of business model, requirements model, analysis model, design model, implementation model, operations model and the retirement model. (I could regroup these into SysML’s business models, architectural models, use case models, behavioral/functional models, requirements models, physical structure/component models and performance/parametric models.) The whole product is never inspected; small parts of it are examined at each inspection.

Work products

Work products of the business model

* Problem statement
	+ describes the customer’s needs,
	+ states the goals of the project,
	+ delineates the scope of the problem,
	+ reports the concept of operations,
	+ describes the stakeholders,
	+ lists the deliverables and
	+ presents the key decisions that must be made.
* Concept of operations (ConOps)
* Context model (an object or data flow diagram) showing how the system fits into its overall environment
* High-level business requirements (essential use cases)
* Domain model (a class diagram) depicting major business classes
* Business process model (an activity diagram) depicting a high-level overview of the business process
* Glossary

Work products of the requirements model

* Operation concept description (OCD)
* System requirements specification (SRS)
* Requirements traceability
* Interface requirements specification
* Preliminary risk analysis
* Use case models for high-priority, high-risk items
* Technical performance measures
* Alternative concepts
* Incipient architectural description
* Process assets library
* Financial analysis
* Business case
* Glossary

Work products of the analysis model

* Elaborations of entities of requirements model
* Analysis packages composed of
	+ high-level use cases
	+ analysis classes
	+ interface definitions
* Use case realizations with
	+ textual descriptions
	+ use case diagrams
	+ class diagrams
	+ activity diagrams
* Special Requirements sections containing
	+ formal shall statement functional requirements
	+ identification of the nonfunctional requirements
* Supplemental entities
	+ functional flow block diagrams
	+ object (context) diagrams
* The analysis model is a static, because it shows objects and their interrelationships, but not time-dependent dynamics.
* Tradeoff studies and key decisions
* Candidate architecture
* Risk analysis
* Financial analysis
* Business case
* Lessons learned
* Glossary

Work products of the design model

* Contains subsystems composed of
	+ design classes
	+ use case realizations
		- textual descriptions
		- class diagrams
		- sequence diagrams
	+ interfaces
		- interfaces of the system and its environment
		- interactions when components of the system are integrated together
		- feedback loops within the system and with its environment
* Special Requirements sections contain
	+ formal shall statement functional requirements
	+ nonfunctional (performance) requirements
* Supplemental entities
	+ state machine diagrams
	+ block definition diagrams
	+ parametric diagrams
* The design model is dynamic because, using sequence diagrams and state machine diagrams, it shows the timing of the functions being performed and the messages being passed
* Tradeoff studies and key decisions
* Architectural description
* Risk analysis
* Lessons learned
* Glossary

Work products of the implementation model

* Contains
	+ components
	+ interfaces
	+ integration build plan
	+ unit test cases and procedures
* Supplemental entities
	+ dependencies between components
* Test plans There are three common techniques for testing systems
	+ test vectors (input/output behavior) and test procedures
	+ system experiments (state-based)
	+ instances of use cases (scenario-based)
	+ State-based testing is the best.

The operations model (usually a computer simulation) should be built from the implementation model. It should reflect the structure of the system as it was actually built. It will be used to manage and improve the operational system. It will be updated anytime the operational system is changed. Most importantly, it will used to help with retirement of the system. An activity diagram will be used to show the workflows, that is who does which tasks during the operations phase.

**Primary Actors**

The formal inspection typically involves an Inspection Team consisting of three to eight members who cover the following roles.

The **Moderator** leads the inspection, schedules meetings, distributes inspection materials, controls the meetings, reports inspection results and follows up on rework issues. Moderators should be trained in how to conduct inspections. The risk or quality assurance managers often serve in this role.

The **Author/Designer c**reates and/or maintains the work products being inspected. The Author/Designer answers questions asked about the work products during the inspection, looks for defects and fixes defects. The Author/Designer cannot serve as Moderator, Reader or recorder.

During the meeting, the **Reader** leads the Inspection Team through the work products being inspected, interprets sections of the artifact by paraphrase and highlights important parts**.**

The **Recorder** classifies and records defects and issues raised during the inspection. The Moderator might perform this role in a small Inspection Team.

The **Inspector** attempts to find errors in the work products. This role can be filled by one or several people. However, all participants act as inspectors, in addition to any other responsibilities. The following may make good inspectors: the person who wrote the specification for the work products being inspected; the people responsible for implementing, testing or maintaining the work product; a quality assurance representative; a representative of the user community; and someone who is not involved in the project but has infinite experience and perfect wisdom.

**Secondary Actors:** Process Asset Library, PAL

**Inspection activities**

**Planning**

The Moderator selects the Inspection Team, assigns roles, obtains work products to be inspected from the Author/Designer, and distributes them and other relevant documents to the Inspection Team two or three days prior to the inspection. The Moderator confirms that the material meets the entry criteria.

**Overview meeting**

This meeting gives the Author/Designer an opportunity to describe the important features of the work products to the Inspection Team.

**Preparation**

Each participant must examine the work products prior to the actual inspection meeting, independently noting any defects found or issues to be raised. Typically, this will take two hours for each participant. The amount of time each person spent will be recorded. Most of the errors found during inspections are identified during this preparation step. The work products should be compared against any predecessor documents or standards to assess completeness and correctness.

**Inspection meeting**

The Moderator and the Reader lead the team through the work products. At the beginning of the meeting, if the Moderator determines that reviewers are missing, that the reviewers did not spend sufficient time reviewing the material or that too much material had been scheduled for inspection, then the meeting should be rescheduled. Michael Fagan recommends inspecting about 130 lines of source code or requirements per hour. The discussion is usually structured as a round robin where each member comments on each issue. During the discussion, all inspectors can report defects or raise other issues, which are documented by the recorder. The Author/Designer can ask for clarification on points raised but is not allowed to defend or explain any defects. Make sure that everyone has the same version of the documents, because the Author/Designer is likely to have a newer version. The meeting should last no more than two hours.

**Rework**

The Author/Designer is responsible for resolving all issues raised during the inspection. This does not mean making every change that was suggested, but an explicit decision must be made about how each issue or defect will be dealt with.

**Follow-up**

The Moderator is responsible for following up with the Author/Designer to verify that the defects fixed in the rework phase have been performed properly. The Moderator should ensure that exit criteria have been satisfied.

**Process to Improve the Process**

All systems should have a Process to Improve the Process (PtItP). The inspection process can be a part of the PtItP. It can reveal kinds of defects being created and process changes that can be made to prevent them. This causal analysis can lead to improved quality on future work by helping to avoid making the same mistakes in the future.

Alternative 8, Requirements Discovery Process



Tradeoff Study Matrix for Diogenes

This may be an old version. Check the Excel file on our class web site for the latest version.

|  |
| --- |
| Tradeoff Study Matrix for Diogenes using a Sum Combining Function and semirelative sensitivity functions. It answers the question, “What is the best architecture to use for a search for unintended consequences process?” |
| Criteria | Wt. | Norm. Criteria Weights | Norm Sub Criteria Weights | Alt 1 Do Nothing, SIMILAR | Alt 2 Risk Analysis | Alt 3 Cause and Effect Analysis | Alt 4 Tradeoff Studies | Alt 5 Test Plan | Alt 6 Sensitivity Analysis | Alt 7 Formal Inspections | Alt 8 Requirements Discovery |
| **number of alternatives, m =** | 8 |   |   | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc | Sc | Wt×Sc |
| **Performance** | 8 | 0.40 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|  Ease of Use | 8 |   | 0.27 | 0.9 | 0.24 | 0.3 | 0.08 | 0.5 | 0.13 | 0.4 | 0.11 | 0.3 | 0.08 | 0.1 | 0.03 | 0.8 | 0.21 | 0.5 | 0.13 |
|  Looks Forward | 10 |   | 0.33 | 0.8 | 0.27 | 1.0 | 0.33 | 0.0 | 0.00 | 0.6 | 0.20 | 1.0 | 0.33 | 0.2 | 0.07 | 0.6 | 0.20 | 0.8 | 0.27 |
|  Has Tools | 4 |   | 0.13 | 0.2 | 0.03 | 0.9 | 0.12 | 0.8 | 0.11 | 0.2 | 0.03 | 0.1 | 0.01 | 0.7 | 0.09 | 0.4 | 0.05 | 0.6 | 0.08 |
|  Inside or Outside | 8 |   | 0.27 | 0.8 | 0.21 | 0.0 | 0.00 | 0.4 | 0.11 | 0.6 | 0.16 | 0.0 | 0.00 | 0.2 | 0.05 | 0.6 | 0.16 | 0.0 | 0.00 |
| **Cost** | 3 | 0.15 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|  Total Lifecycle Cost | 4 |   | 0.44 | 0.8 | 0.36 | 0.5 | 0.22 | 0.5 | 0.22 | 0.5 | 0.22 | 0.5 | 0.22 | 0.5 | 0.22 | 0.5 | 0.22 | 0.5 | 0.22 |
|  Operating Cost | 5 |   | 0.56 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 | 0.5 | 0.28 |
| **Company Policy** | 9 | 0.45 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|  BiST | 10 |   | 0.43 | 0.2 | 0.09 | 0.4 | 0.17 | 0.2 | 0.09 | 0.8 | 0.35 | 0.0 | 0.00 | 0.8 | 0.35 | 0.3 | 0.13 | 0.5 | 0.22 |
|  Reusability | 6 |   | 0.26 | 0.9 | 0.23 | 0.4 | 0.10 | 0.4 | 0.10 | 0.8 | 0.21 | 0.0 | 0.00 | 0.8 | 0.21 | 0.9 | 0.23 | 0.7 | 0.18 |
|  Vendor Evaluation | 7 |   | 0.30 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 | 0.5 | 0.15 |
| **Alternative Rating** |   |   |   |   | **0.61** |  | **0.48** |  | **0.37** |   | **0.59** |  | **0.31** |  | **0.49** |  | **0.56** |  | **0.52** |

The Tradeoff Study Matrix above shows that the recommended alternative is the Do Nothing alternative, the SIMILAR process. The complete tradeoff study and sensitivity analysis is located at http://www.sie.arizona.edu/sysengr/sie554/unintended/index.html

The values for the cells were derived by a panel of domain experts using the Delphi method. The values came from manufacturers’ data, peer reviewed journal papers and Internet web sites. A value was derived for each alternative for each criterion. This value was put into the scoring function for that criterion and the resulting score (Sc) was put into the matrix. *{I do not have scoring functions yet.}* Each score was then multiplied by its corresponding normalized weight (Wt) and these products were summed in each column to produce the Alternative Ratings.

This tradeoff study answered the question “What is the best architecture to use for a search for UiCs process?” by stating that the preferred alternative is the Do Nothing alternative, the SIMILAR process.

Each number in a tradeoff study matrix should have a verbal explanation. For example, consider the evaluation criterion *Ease of Use*. The system should be intuitive to use. There should not be a long learning curve. The system should hide the mathematics from the user. Low complexity is desired.

(1) The SIMILAR process has been used by millions of lay people for eons. So it gets a score of 0.9.

(2) For a risk analyses process (as used in industry), the risk register, risk plots and risk waterfall are easy to use, but finding the risks and gathering quantitative data are difficult. So it gets a score of 0.3.

(3) A cause and effect process is easy to use: but using it well takes a lot of experience. It gets a score of 0.5.

(4) A tradeoff study matrix is easy to construct: but avoiding mistakes is difficult. The sensitivity analyses part of a tradeoff study is difficult. It gets a score of 0.4.

(5) Traditionally, a comprehensive test plan was difficult to create: but now deriving one from use cases is easy. It gets a score of 0.3.

(6) A sensitivity analysis is used to evaluate the sensitivity of SystemZ to its inputs and parameters. Understanding the mathematics is difficult for lay people. It gets a score of 0.1.

(7) The formal inspection process was well designed. It gets a score of 0.8.

(8) The requirements discovery process requires a knowledgeable person. It gets a score of 0.5.

The values were originally estimated by Bahill, September 20, 2010. The most recent revision was done on November 6.

The second performance evaluation criterion is *Looks Forward.* The search for UiCs process must look forward in time. Admittedly, each tool could be modified to look forward or backward in time. But for tools that have been in use for decades, there is so much literature and knowledge available, that switching would be confusing.

(1) The SIMILAR process looks forward in time, backward in time and at the present time. Therefore, it gets a score of 0.8.

(2) A risk analyses process looks only forward in time. It gets a score of 1.0.

(3) A cause and effect process is used after there is a failure: it looks backward in time. It gets a score of 0.0.

(4) A tradeoff study process is used to make decisions involving the design of SystemZ. It looks at present and future times. Some evaluation criteria could be potential UiCs. It gets a score of 0.6.

(5) A comprehensive test plan process is used to test SystemZ and its requirements. It looks only forward in time. It gets a score of 1.0.

(6) A sensitivity analysis process is used to evaluate the sensitivity of SystemZ to its inputs and parameters. It computes sensitivities at the present time. But it can reveal unplanned sensitivities to mundane parameters and interactions. It gets a score of 0.2.

(7) The formal inspection process was designed to look in the present time: but there is no history that impedes it from looking forward. It gets a score of 0.6.

(8) The requirements discovery process was designed to look at present and future times. It gets a score of 0.8.

The values were originally estimated by Bahill, September 8, 2010. The most recent revision was done on November 6..

Consider the criterion *Has Tools.* The system should have tools to help the systems engineer to discover UiCs. Brainstorming is an example of a primitive tool. Prioritization is a simple tool. A fishbone diagram is a sophisticated tool.

(1) The SIMILAR process uses brainstorming, but it has nothing special, just a flow diagram. So it gets a score of 0.2.

(2) The risk analyses process has the risk register, risk plots, risk waterfall and HHM. So it gets a score of 0.9.

(3) A cause and effect process has cause and effect diagrams and Ishikawa fishbone diagrams. It gets a score of 0.8.

(4) A tradeoff study matrix and the sensitivity matrix are the tools. But these tools might not help us to find UiCs. It gets a score of 0.2.

(5) A comprehensive test plan uses test vectors and test procedures (input/output behavior), system experiments (state-based) and instances of use cases (scenario-based). But these tools will not help us find UiCs. It gets a score of 0.1.

(6) A sensitivity analysis uses partial derivatives and sensitivity functions. Unplanned excessive sensitivity to any parameter or sensitivity to interactions might flag UiCs. It gets a score of 0.7.

(7) The formal inspection process has well defined roles and activities. It can use a multitude of commercially available tools, but doesn’t have many tools of its own. It gets a score of 0.4.

(8) The requirements discovery process is well defined. MS Office tools and commercial requirements database management tools such as DOORS will suffice for most activities. It gets a score of 0.6.

The values were originally estimated by Bahill, September 20, 2010. The most recent revision was done on November 6.

Finally, let us consider the performance evaluation criterion *Inside or Outside*, whichevaluates whether the tool looks for events (causes, failures, sensitivities, etc.) that are inside of the system being designed (SystemZ) or outside of SystemZ. The search for UiCs process must look outside. Admittedly, each tool could be modified to look inside or outside of SystemZ. But for tools that have been in use for decades, there is so much literature and knowledge available, that switching would be confusing.

(1) The SIMILAR process does not care whether the event is inside or outside of SystemZ so it gets a score of 0.8.

(2) A risk analyses process (as used in industry) looks only for risks to SystemZ or its primary actors: so it gets a score of 0.0.

(3) A cause and effect process is used to look for the root causes of failures of SystemZ. It only looks for failures of SystemZ, but it can invoke causes that are outside of SystemZ. It gets a score of 0.4.

(4) A tradeoff study process is used to make decisions involving the design of SystemZ. However, we could create evaluation criteria that would look, for example, at harm to the environment. It gets a score of 0.6.

(5) A comprehensive test plan process is used to test SystemZ and its requirements. It would be hard to use it to look for outside entities. It gets a score of 0.0.

(6) A sensitivity analysis process is used to evaluate the sensitivity of SystemZ to its inputs and parameters. But the performance function could contain outside parameters. It gets a score of 0.2.

(7) The formal inspection process was designed to inspect SystemZ: but there is no historical baggage that impedes it from looking outside. It gets a score of 0.6.

(8) The requirements discovery process was designed to look at SystemZ. It gets a score of 0.0.

The values were originally estimated by Bahill, September 8, 2010. . The most recent revision was done on November 6.

One of the Company Policy criteria is the existence of *Built-in Self-Test*: BiST could be passive, where the BiST system monitors outputs and check points and displays status: or BiST could be active, where the BiST system generates signals and applies them to the system inputs.

(1) The SIMILAR process could use peer review. It gets a score of 0.2.

(2) A risk analyses process could check the numbers for validity, completeness and consistency. It gets a score of 0.4.

(3) A cause and effect process could use peer review. It gets a score of 0.2.

(4) A tradeoff study process could check the numbers for validity, completeness and consistency. Excel has lots of built in tests. It gets a score of 0.8.

(5) I have never heard of a test plan to test the test plan. It gets a score of 0.0.

(6) Examining the symmetry and regularity of the equations in a sensitivity analysis is a good error detection metric. My examples have two sensitivity matrices computed with different data and formulae, but they contains the same values. It gets a score of 0.8.

(7) The formal inspection process uses management oversight. The metrics analysis group will analyze the metrics. It gets a score of 0.3.

(8) Examining the regularity of the grammar and structure of the requirements could be used for BiST. It gets a score of 0.5.

The values were originally estimated by Bahill, September 20, 2010. The most recent revision was done on November 6.

One of the Company Policy criteria is *Reusability*: It is company policy that non-developmental products be considered the primary solution to addressing customer needs with customized implementations being secondary solutions. In the tradeoff study, each alternative will be given points for use of COTS products and will be deducted points for use of custom-made products. Each alternative will be given points if the analysis and results are likely to be reused in different environments.

(1) The SIMILAR process has been used and reused by millions of lay people for eons; and it is free. So it gets a score of 0.9.

(2) There are COTS packages for risk analyses. However, outputs of a risk analysis are marginally usable in future analyses. It gets a score of 0.4.

(3) There are COTS packages for cause and effect analyses. However, the outputs of a cause and effect analysis would not be usable in future analyses. It gets a score of 0.4.

(4) There are no excellent COTS packages for tradeoff studies; however, everyone can use my examples and templates. The artifacts are extremely reusable in future studies. It gets a score of 0.8.

(5) Test plans and procedures are extremely specific. It gets a score of 0.0.

(6) There are no good COTS packages for sensitivity analyses; however, everyone can use my examples and templates. My Excel spreadsheets are very reusable in future studies. It gets a score of 0.8.

(7) The formal inspection process is used over and over again. It gets a score of 0.9.

(8) Most requirement systems use DOORS. The requirements themselves are often reused. It gets a score of 0.7.

The values were originally estimated by Bahill, September 20, 2010. The most recent revision was done on November 6.

When filling in a tradeoff study matrix, it is important to pay attention to common mental mistakes (Smith, Son, Piattelli-Palmarini and Bahill 2007). For example, the criteria Total Lifecycle Cost and Operating Cost are dependent and evaluation criteria should be independent.

The sensitivity analysis (Smith, Szidarovszky, Karnavas and Bahill, 2008) performed on this tradeoff study showed that the most important category weight was that for *Company Policy*, the most important subcategory weights were for *Looks Forward* and *BiST*, and the most important scores were for *BiST* for alternatives 4 and 6, and *Looks Forward* for alternatives 2 and 5. Therefore, we spent extra time discussing amongst ourselves and with top management, whether *Performance*, *Cost* or *Company Policy* should have the highest weight. We expended extra resources consulting with management and the customer about the relative weights of the subcriteria. We did extra research, analysis and modeling to get scores that are more reliable for *BiST* for alternatives 4 and 6, and *Looks Forward* for alternatives 2 and 5.

|  |  |  |  |
| --- | --- | --- | --- |
| Sensitivity rank | Criteria | reliability | uncertainty |
| 1 | Company Policy weight | 8 | 8 |
| 2 | Looks forward weight | 9 | 9 |
| 3 | BiST weight | 9 | 9 |
| 4 | BiST score for Tradeoff Studies | 4 | 4 |
| 4 | BiST score for Sensitivity Analysis | 4 | 4 |
| 6 | Looks Forward score for Risk Analysis | 9 | 9 |
| 6 | Looks Forward score for Test Plan | 9 | 9 |