Technical Report
CMU/SEI-94-TR-14
ESC-TR-94-014

A Construct for Describing Software Development Risks

David P. Gluch

July 1994

Technical Report
CMU/SEI-94-TR-14
ESC-TR-94-014
July 1994

A Construct for Describing Software Development Risks

David P. Gluch

Team Risk Management Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the

SEI Joint Program Office
HQ ESC/AXS
5 Eglin Street
Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Thomas R. Miller, Lt Col, USAF
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-95-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 52.227-7013.

This document is available through Research Access, Inc., 800 Vinial Street, Pittsburgh, PA 15212.
Phone: 1-800-685-6510. FAX: (412) 321-2994. RAI also maintains a World Wide Web home page. The URL is http://www.rai.com

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145. Phone: (703) 274-7633.

Table of Contents

List of Figures

A Construct for Describing Software Development Risks

Abstract: This report establishes a representation of software risk wherein the risks associated with software-dependent development programs are defined as distinct, manageable risk entities. The risk entities and their descriptive statements of risk are based upon a Condition-Transition-Consequence (CTC) construct. The CTC construct arises out of a systems representation, where time and value are identified as fundamental to the concept of risk. The CTC construct is also shown to provide a common representation for both program risks and program tasks and to fit into a heuristic framework for identifying risks within software-dependent development programs. Examples of risks are used to demonstrate that the approach facilitates the management of risk as an integral part of routine program management.

1 Introduction

Dealing with risk is an important aspect of decision making in industry, government, and academic endeavors [Haimes 89]. Risk analysis and risk management have been applied extensively in considering the operational and safety aspects of large complex systems which may adversely affect the health and safety of society [Bell 89], [Henley 92]. Risk management has also been identified as an important, but often lacking, element in effective decision making for and routine management of software engineering and development programs [Boehm 91], [SEI 92], [Charette 90], [Kirkpatrick 92], [Haimes 91], [Chittister 93].

This report presents a formalism for describing the risks associated with software-dependent development programs such that risks can be managed as an integral part of routine program management activities. The formalism is based upon a structured format for a statement of risk: the Condition-Transition-Consequence (CTC) format. A CTC-based approach for risk identification and management establishes a distinct risk entity that embodies the essence of the concept of risk and evolves throughout all phases of the life cycle of a program.

In Section 2 of the report, the concept of risk as a distinct entity is discussed and time and value are identified as fundamental to risk. A state-space representation for software-dependent development programs and its connection to risk are presented in Section 3. In Section 4, the CTC construct is defined and example CTC statements of risk are presented. The application and implications of the CTC approach as well as some extensions to the basic concept are presented in Section 5. In Section 6, CTC risks and program tasks (activities) are compared. Section 7 introduces a heuristic for risk identification – the CTC triangle. In Section 8, the CTC approach is compared and contrasted with related risk terminology and risk concepts. Section 9 summarizes the CTC approach.

2 The Concept of a Risk Entity

Current definitions of risk, as a noun, include [Houghton 85]:

1 The possibility of suffering harm or loss; danger
2 A factor, element, or course involving uncertain danger; hazard
3.a The danger or probability of loss to an insurer
3.b The amount an insurance company stands to lose
3.c A person or thing considered with respect to the possibility of loss to an insurer: a poor risk.

A noteworthy example of a more general use of the term risk is in the field of Operations Research. In Operations Research, the concept of decision under risk describes situations where there is a probability associated with an outcome or choice, regardless of the nature of the outcome [Taha 87]. But for the most part the term is used as reflected in the following definitions [Gove 81]:

1 the possibility of loss, injury, disadvantage, or destruction
2 someone or something that creates or suggests a hazard or adverse chance
3 the chance of loss or the perils to the subject matter of insurance covered by a contract
4 the product of the amount that may be lost and the probability of losing it

In the context of software engineering and development, risk can be defined as the possibility of suffering a diminished level of success (loss) within a software-dependent development program. This prospect of loss is such that the application of the selected theories, principles, or techniques may fail to yield the right software product [SEI 92].

2.1 Value and Risk

The potential loss to the program and specifically the association of risk with the program involves a value judgment on the potential impact of risks on the program. The terms loss, danger, hazard, and harm, all of which reflect a negative perception, involve at least a relative assessment of value [Rowe 88].

Many attributes of a program can be used to characterize value in the context of software-dependent development programs. For example, a sense of value may be expressed in the form of:

- customer satisfaction
- software execution speed
- software code size
- date of delivery
- number of software bugs
- user friendliness

Quantitative or qualitative attributes may be used to represent “value” and the sense of value associated with the specific measure of the attribute is based upon the perception, the subjective interpretation, of the worth of these attributes to the customer, user, or other individual or organization critical to the success of the program. Clearly the sense of value makes the definition of what is a risk a very subjective one, but this sense of value, more specifically negative value, is fundamental to the concept of risk.

2.2 Time, Uncertainty, and Risk

It is clear from the various definitions of risk that uncertainty expressed as possibility or probability is involved with risk. As Rowe points out, uncertainty involves both descriptive and measurement uncertainties [Rowe 88]. In addition, as noted by [Charette 90] the nonlinear, nondeterministic character of the dynamics of the environment also contribute to uncertainty. Uncertainty arises not simply from the inability to measure or describe exactly the circumstances, etc., associated with risk; but collectively from the kinematic and dynamic characteristics of the environment as the world evolves in time.

The interrelationship of uncertainty and time is evidenced in the uncertainty associated with risk, in that this uncertainty reflects the uncertainty regarding future events [Kloman 90], [Rowe 88]. While the prospects for the future may be uncertain due to a multiplicity of factors, the passage of time is required to realize the outcome of the circumstances that exist now. Uncertainty is inherent in the evolution of the environment and the temporal aspect of risk within an uncertain environment is identified here as fundamental to the concept of risk.

Given the perspective outlined above, risk is a concept that embodies a sense of value and a sense of time; and these two attributes, *value* and *time*, are identified as the fundamental characteristics in a representation for identifying and managing software technical risks and for defining a risk entity within the context of a software-dependent engineering and development program.

2.3 Risk as a Manageable Entity

While risk is an important operational consideration, it is abstract and difficult to grasp. Pragmatically then, to enable the management of risk, specific tangible exemplars of risk (identified here as risk entities) are established. From this perspective, a *risk* will be used to designate a tangible entity that embodies the concept of risk and captures the essential elements of the concept. In particular, this risk entity is characterized by both descriptive and measurable attributes that capture the essential elements of risk, and that relate, directly or indirectly, to factors critical to successful program management, e.g. budget, performance, and schedule.

In this representation the characteristics of uncertainty, probability, impact, etc. often included in a definition of risk [Lowrance 76], [Rowe 88], [Charette 89], [Charette 90], [Kirkpatrick 92], [Kaplan 81] are represented as attributes of the risk entity. A risk, then, is an entity defined by

the values of characteristic attributes associated with risk that evolves through time under the processes, constraints, and uncertainties inherent in the environment. This evolution of a risk entity includes a life cycle of creation, existence, change, and extinction.

Because risk is fundamentally subjective, the reification of risk as a risk entity provides a basis for a software-dependent development program to deal with risk more objectively. The approach can be viewed as transforming a perception of risk on the part of any individual (an intangible) into a distinct risk entity that can be described and measured. The process of going from the perception of risk to its representation as a risk entity is defined as risk identification. With the explicit identification of a risk entity and its descriptive and measurable attributes, risks can be analyzed, tracked, and controlled as part of the routine program management practices.

3 A Program as a System

In order to define a structure and a representation for risk entities, both a model and a modeling language [Levy 93] for a software-development program, project, or engineering development effort are identified. The model is based upon a state-space representation, similar to the systems models used in physics, systems analysis, and related fields and as employed in artificial intelligence approaches to problem solving and planning [Simon 83]. Specifically, a software-dependent development program is represented as a complex system within an n-dimensional space, the program space. This approach is not unique; others have used a systems model to represent programs and associated risks [Charette 90]. The perspective presented here is intended to provide a foundation for the representation of risk as risk entities.

The program space is visualized as an n-dimensional space with dimensions that are defined based upon the perception of the individuals involved in the program. A reference frame is defined as the axes set for the program space and is used for describing the program as shown in Figure 3-1. Each axis is a descriptive characteristic of the program and has associated with it a value[1] scale (measure[2]) along that axis. One of the axes of the reference frame is the time axis. For each characteristic (axis of the frame) an individual program has a measure (value) on that axis, resulting in a characteristic-measure pair, as shown in Figure 3-1. These axes can also be viewed as attribute-value pairs in the context of a semantic frame representation whose form is flexible and depends upon the context of the application. In this case, a frame would be a set of related axes, with each set or frame related in a hierarchical network [Basalou 92].

Within the context of the program space, the program is represented as a system that is described in terms of measures along each axis of a reference frame, whose axes are defined by program attributes. The program itself is represented as a point in the space; due to the uncertainties associated with the attributes, the environment, and the reference frame definition itself, the program is represented as a fuzzy point in space, which is arbitrarily shaped and larger than a single point. The characteristics which define an axis may entail qualitative or quantitative measures. For example, axes of the frame may include time, customer satisfaction, performance, cost, maintainability, or any characteristic that may be used to describe the program.

The system can also be described in terms of a state vector [Charette 90, p. 69.], $\boldsymbol{V}(\boldsymbol{x},t)$, where the vector $\boldsymbol{x}$ has the components x_i, i = 1 to n. The value x_i represents the magnitude of the variable on the x_i axis and t is the magnitude on the time axis. Graphically, the point in the space for the system can be visualized as the end point at the tip of a vector which has its origin at the center of the reference frame, as shown in Figure 3-1.

---

1. Value is used here as in the sense of *magnitude of* or *measure of* or *size of*.

2. Measure is used here to mean some *qualitative or quantitative assessment of the characteristic*, e.g. large, medium, small or 5, 10.5, etc., as appropriate for the characteristic or the level of description or analysis desired.

At any given time, the reference frame consists of an instantaneous set of characteristics which can be used to represent the program and associated environment. Any one axis may change in its scale, importance, range, etc. and may be abandoned in the representation at some future time. The reference frame as well as the system state itself is dynamic.

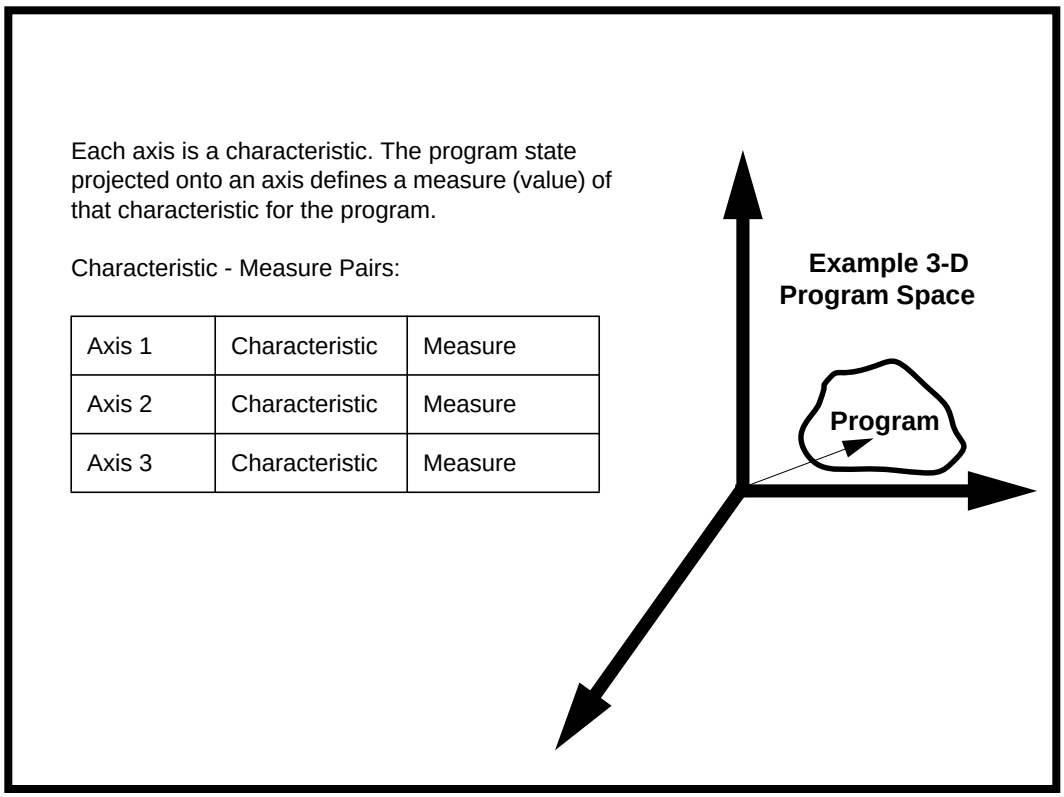


Figure 3-1: The Program Represented as a "Point" in the Program Space

The subjective character of the reference frame is evidenced in the freedom of choice of axes (the characteristics that form the frame), the number of axes, and the measures assigned to the system on each of the axes of the frame. The axes of a program frame range over a diverse set of characteristics that describe the program. Table 1 is an example of a few of the possible choices for characteristics and associated measures.

Axis (Characteristic)	Measure (value)
Program Staff Size	150
Number of Lines of Code	1.5 million
Requirements Stability	low
Language	Ada
Type of Contract	Fixed Price
Program Duration	36 months
Development Model	Spiral

Table 1: Example Program Reference Frame Characteristic-Measure Pairs

Since a reference frame is subjective, there is a separate program reference frame for each of the individual perspectives represented in the program. When these individual perspectives are shared, the prospect of a program-wide reference frame[1] exists and a formal representation (joint interpretation) of the program may be developed. This mutually agreed-upon reference frame can be represented in functional specifications, requirements documents, program plans, etc. The global program reference frame is dynamic in character and transcends individual perspectives by being modified only through mutual agreement.

Through the identification of a software-dependent development program as a system within a dynamic reference frame, the characteristics of the evolution of the program through its life cycle can be described in terms of the kinematics and dynamics of both the system and the frame of reference. Uncertainty is inherent in the dynamic processes which characterize both the program's evolution and the evolution of the reference frame.

3.1 Risk and the Value-Time System

To focus on the risk aspects of the program, a representation of a system (program or development project) is projected onto a value versus time axes set, a value-time frame. This projection is shown in Figure 3-2. A value axis can be any single or collective "value" characteristic of the program. Given the subjective and fluid nature of the reference frame, the specific value axis, measure, etc. may change through time.

1. The second level and deeper levels are not considered here, i.e. where each individual interprets, perhaps differently, the program-wide reference frame based upon their individual perspective.

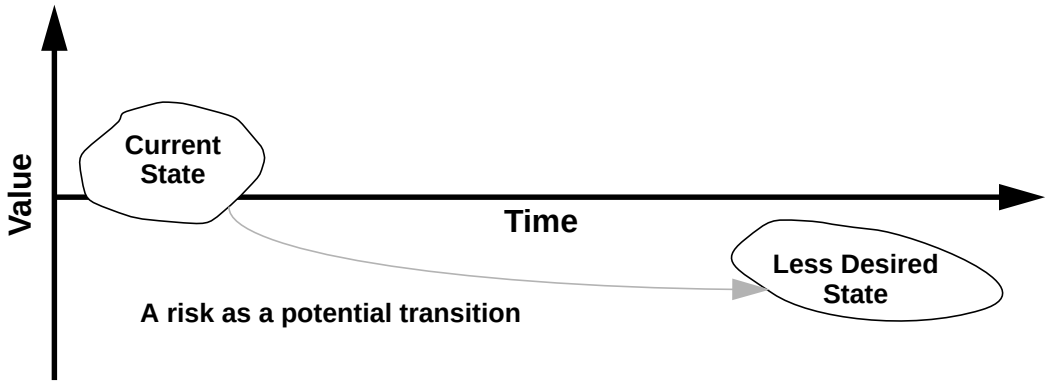


Figure 3-2: Value-Time Frame of Reference

A value axis is divided into two parts based upon a qualitative criterion of acceptable/desired or unacceptable/undesired. Any perceived measure of worth relating to the program, e.g., customer satisfaction, quality, etc. can define a value axis. The state of the system is represented by a fuzzy point to reflect the uncertainties associated with the actual magnitudes of the descriptive variables for the system.

In addition, the boundary between acceptable/desired and unacceptable/undesired is subjective and dynamic. The frame, characteristics, characteristic measures, and the boundary will evolve and change throughout the life of the program.

3.2 Risk as a Transition

As noted [Charette 90], given the state-space representation, risk can be associated with a *potential* state transition in the space. (Specifically, Charette identifies a risk as “the likelihood of making a particular state transition”; and associates with the risk, a description of the undesirable consequence of the transition.) In the value-time plane, then, a program risk can be thought of as a *potential* transition between the current state of the system (program) and an undesired end state. Risk, geometrically, is a trajectory in the plane; risk identification in this representation is the recognition of a potential transition path (trajectory) that carries the state of the system from its current state to an undesired new state.

The recognition of a possible transition need only occur from the perspective of one individual associated with the program. On some larger or absolute scale, the state is not undesired but what is significant to the identification of risk is that at least one individual has the perception (concern) that a transition may occur to a final state is of less “value” than the state that was planned to exist.

While the projection of the system onto a two-dimensional value-time frame simplifies the representation, in general, a risk can be identified in the n-dimensional frame as a potential transition where:

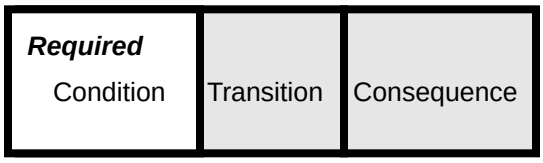
- The system is a fuzzy point in the space, and
- A risk is a potential transition, a trajectory through the space, from the current state of the system to some other (undesired) state.

It is important to realize that the representation, wherein the program is a state within the program space and a risk is a potential path in that space, is an instantaneous view and is transient in character. This representation is a snapshot of the perceptions of the system at that instant. All of the elements that describe the program, their measures, and their perceived value to the program and the risks will likely change.

In general, a risk can be modeled as a transition in a multidimensional, subjective, and dynamic program space, and in particular, as a projection of that transition onto a *value* versus *time* axes set. The value-time subspace is used as a basis for a subjective representation of a risk, one expressed in terms of the fundamental components of value and time.

4 CTC Representation

By connecting the concept of a risk entity with the representation of a risk as a potential transition of a system within a dynamic multidimensional program space, a risk can be expressed as a construct consisting of a description of the initial state of the system (condition), the potential evolution of the system (transition), and the potential final state (a less desirable state) of the system (consequence). This representation of a risk entity, the Condition-Transition-Consequence (CTC) construct for a risk, is shown in Figure 4-1.



***Condition** is a description of the current conditions prompting concern.*
***Transition** is the part that involves change (time).*
***Consequence** is a description of the potential outcome.*

Figure 4-1: A CTC Risk Construct

Given that time and value are fundamental to risk, the minimum required to identify a risk (a risk entity in the form of a CTC representation) is the statement of the conditions coupled with an expression of concern about the potential consequences. The undesirable end state need not be explicitly defined and the details of the transition, which will involve considerations of the dynamics of the system, need not be specified in the identification process. This minimal statement is sufficient for risk identification in the CTC representation of risk and provides the basis to initiate subsequent steps in the management process.

4.1 Format for Risk

Given the CTC construct, a format for a statement of a risk can be formulated as:

- **Given that** *condition* **then there is concern that (possibly**) *consequence.*

The **Given** (condition) portion of the statement is intended to establish the conditions that exist now that have prompted the concern regarding the program and have engendered the perception of the prospect of diminished success (loss).

The CTC construct as the basis for a statement of a risk and the treatment of risk as a distinct entity makes risk manageable. Specifically, the circumstances that exist within a program, the condition portion of the CTC format, can be dealt with in an attempt to mitigate, alleviate, and monitor risks, and thus enable risk management. A clear definition of where the program is

now, i.e., knowing where things stand, is vitally important in developing a plan to change the conditions. Therefore, the condition component of the statement of risk, a description of the current conditions, is the minimum required to define a risk.

> A minimal statement of *a risk i*n the CTC sense is defined as the description of the current conditions coupled with an expression (possibly implied) of concern (potential loss) regarding those conditions.

The expression of concern may be embodied in a value judgment within the statement of the condition. For example, terms like “lack of”, “inadequate”, “insufficient”, can be used to express the concern more explicitly regarding these conditions. Note that this can be viewed as a value judgment on the transition-consequence, i.e., in the judgment (perception) of the individual there is “less” than required now to do the job, hence there is concern that the program will suffer some diminished level of success.

An explicit value judgment within the condition is not required. For example, a risk may be: *Given the current level of testing of the software then there is concern...*. The implication is that the level as it exists now is not adequate. In general, the consequence portion is considered optional during the identification process and can be expressed as notional.

Overall, the Condition-Transition-Consequence construct defines a risk entity and establishes the CTC format for a statement of a risk as shown in Figure 4-2.

Given that *Condition* then there is concern that (possibly) *Consequence*

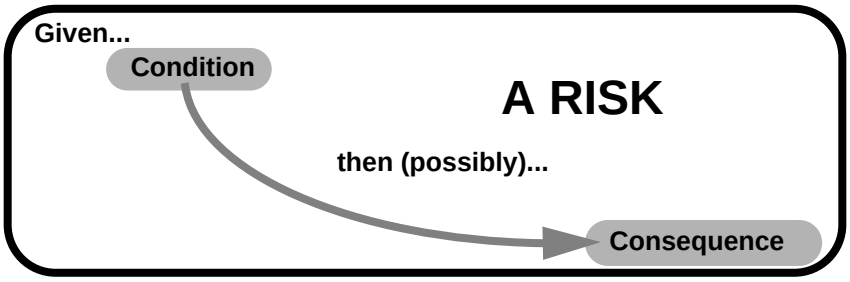


Figure 4-2: The CTC Construct and the Format for a Risk Statement

4.2 The Attributes of a Risk

While the CTC construct provides an explicit form for the statement of a risk as well as a way to view a risk entity, the concept of a CTC risk entity also includes a descriptive attribute-value set. The attributes are characteristics of risk such as probability, impact, risk exposure, time frame information, metrics for tracking and control, as well as administrative information that

provide important details on the risk. A partial but representative set of attributes for a CTC risk is shown in Figure 4-3. Collectively, the attribute set and associated values include all of the relevant detail on the nature of a specific risk entity that is required to fully understand and manage that risk.

One of the key attributes of the CTC representation is the context[1] attribute of the risk. The context consists of a detailed description of the events, circumstances, and inter-relationships that may affect the program; this description is more detailed than can be captured in the basic CTC risk statement. Collectively, the CTC statement of risk and the context provide sufficient information to initiate the analysis and planning processes required to manage the risk.

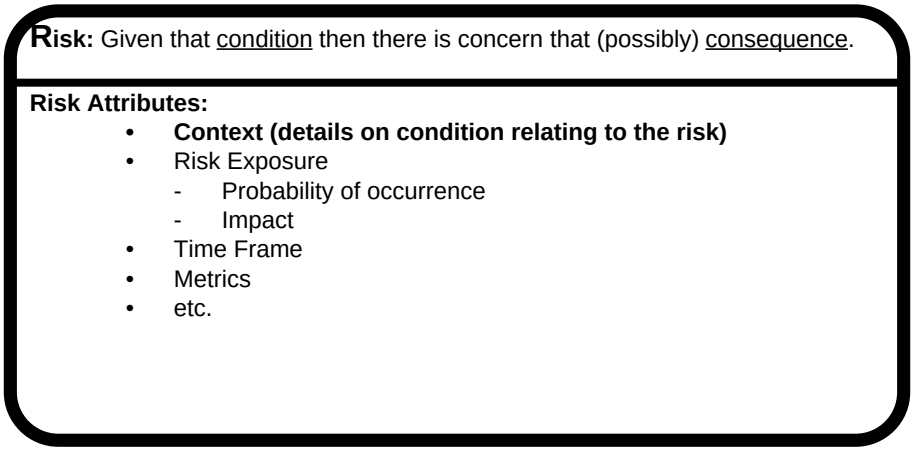


Figure 4-3: An Example CTC Risk Entity with Attributes

An example of the CTC risk statement and associated context is shown below:

> **Given that** the graphical user interface (GUI) must be coded using X Windows and we do not have expertise in X Windows software, **then there is concern that (possibly)** the GUI code will not be completed on time and will be inefficient.
>
> **Context**: The GUI is an important part of the system and we do not have anyone trained in the X Window System. We all have been studying the language but it is complex. Also, only one person in the group has any graphics experience and that is with Windows on the PC.

1. The context of a risk originated as part of the Software Engineering Institute Team Risk Management project efforts. Specifically recognized is the contribution of Ron Higuera in establishing the use of the context in the risk management process.

As can be seen in the example the context provides additional information regarding the condition of the program relative to the nature of the concern and details of the circumstances that have prompted the concern.

5 The Application of CTC

Consider a conversation between members of a software development team during which a comment is made to the effect “I am concerned about getting the integration lab completed on time.” Clearly there is a sense of risk to the program on the part of the individual making the comment. To capture this risk and to provide a basis for managing the risk, the identification using the CTC statement of risk format would involve defining the conditions (circumstances) that have prompted the concern. The approach would involve additional probing to ascertain the basis for the concern and factors that have engendered the concern. This formulation is in contrast to a risk identification approach and representation which would capture a statement to the effect that: “There is a risk of not completing the integration lab on time.”

For this example the CTC risk would be identified as:

- Given that *the Government Furnished Equipment (GFE) has not functioned as planned* then there is concern that (possibly) the integration lab will not be completed on time.

The context would appear as:

> There has been a substantial amount of work required to get the GFE operational and we have not as yet gotten the equipment to function as we had planned. So far it looks like some functions we had anticipated being there are not fully debugged and may not work as specified. Also, the efforts to get the equipment working to date have taken longer and required more personnel than planned.

It may also be that there are multiple conditions (factors) that prompt the concern and (as in the example discussed above) additional risks might be identified. Each new description of circumstance or condition would be identified as a separate risk:

1 Given that *no one is tasked with testing the laboratory equipment* then there is concern that (possibly) the integration lab will not be completed on time.

and

2 Given that *the requirements for the integration lab are not finalized* then there is concern that (possibly) the integration lab will not be completed on time.

and

3 Given that *no plan exists for building the integration lab* then there is concern that (possibly) the integration lab will not be completed on time.

It is also important to realize that the focus is on the condition statement rather than forcing or even encouraging the details of the consequences during any risk identification process. It is more important and expedient to capture the conditions since the basic statement of consequence is captured in the CTC statement and the details relating to impact, probabilities, etc., are captured as part of the context and in the risk attributes. Additionally, details on the nature

of the consequence(s) can be addressed as part of subsequent risk management steps. For example, the collective impact of all of the consequences and their interplay can be addressed as part of the analysis and planning steps.

Within a CTC statement, the core of the risk is identified. From this basic information, program personnel can pursue follow-on activities as part of the process steps of the SEI risk management paradigm [SEI 92]. Specifically, determinations of risk properties (attribute-value pairs) can be readily accomplished in terms of interacting conditions that increase or decrease the probability of potential consequences and whether these conditions reinforce or modify one another, i.e. the interactions of risks can be analyzed based upon the potential interplay of conditions.

It is also important to separate the statement of the conditions from a root-cause analysis process. A root-cause analysis provides additional insight into the nature of the risk and the potential effectiveness of various mitigation strategies. The CTC statement provides a structure for a concise statement of risk and a foundation for subsequent analysis, e.g. root-cause analysis, as part of the later steps in the risk management process.

5.1 A Simplified Notation

In the recording of the statement of a CTC risk, a short-hand notation can be used. Specifically, the words “Given that” can be omitted, and the phrase “then there is concern that (possibly)” can be replaced by a semicolon. For example:

> no one is tasked with testing the laboratory equipment; the integration lab will not be completed on time

or

> no plan exists for integrating the graphical user interface software into the system;

5.2 A CTC Risk and Risk Identification

A CTC-based approach to risk management and specifically to the identification of risk in a software-dependent development program involves a process of probing conditions that exist in a program rather than strictly exploring the potential areas of impact and viewing the reverse chain of causal events. The contrast between the exploration of conditions rather than consequences in risk identification is similar to that of the event tree approach to risk analysis (forward logic) versus the fault tree analysis of risk (backward logic) [Henley 92].

Specifically, in the CTC-based approach, each identified risk entity would be described by a formal CTC statement of risk and an associated context. The statement of risk would include at least a notional sense of the potential consequences. The elaboration of impact, probabilities, etc., would be completed as part of subsequent risk management process steps. In this respect the CTC-based approach is similar to the preliminary hazards analysis process of identifying hazards and their potential consequences [Henley 92] but focuses on the elabora-

tion of the conditions that define the existing state of the program. With the CTC representation of risk, the SEI's continuous risk management paradigm [SEI 92] [Higuera 93] would involve the structuring of the statement of risk with context - defining a risk entity - and the subsequent management of this entity throughout all steps of the paradigm.

5.3 Extensions of the CTC Format

The CTC construct allows for the possibility that a singe condition has a multiplicity of consequences for the program. There are two distinct possibilities: co-occurring and cascade, as shown in Figure 5-1 and Figure 5-2. The first case involves consequences which co-occur in time. The Cascade consequences are serial outcomes, generally causally related.

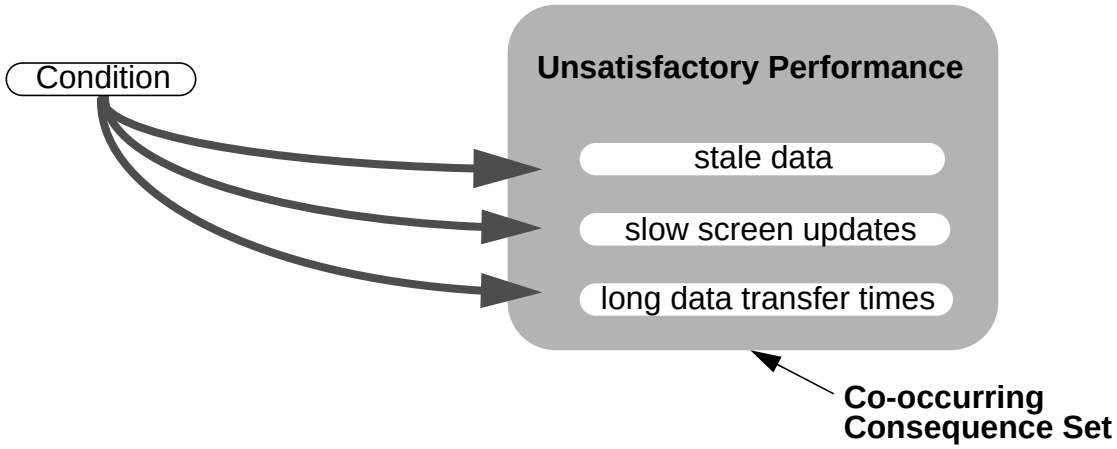


Figure 5-1: Co-occurring Consequences

The statements of risk for these hypothetical cases may be of the form:

- Co-occurring statement of risk

  the code completed so far is 25% of the total planned lines of code but requires 50% of the available time frame; stale data & slow screen updates & long data transfer times

- Cascade statement of risk

  there is a lack of C coding experience on the real-time signal processing code development team; late code & late into test & late into first build

The multiplicity of consequences can be included as part of the scenario attribute of the CTC risk. The scenario set describes in more detail the potential evolution of the program based upon the CTC condition. Scenarios provide detail on the transition aspects of a CTC risk entity. The scenario(s) would be generated and modified, as needed, throughout the risk management process as part of detailed analysis, planning, tracking, and control.

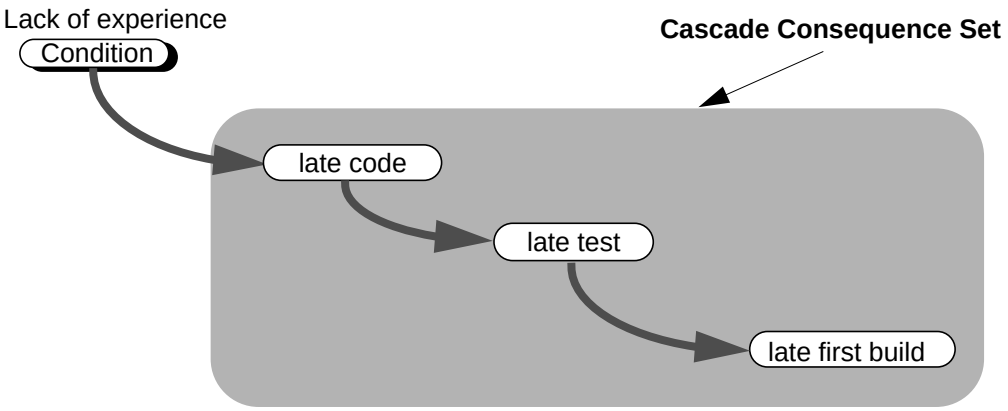


Figure 5-2: Cascade Consequences

A single CTC risk can be decomposed into more detailed risks to the extent that a general condition has a multiplicity of individual characteristics. In the CTC representation, these distinct risks can share common consequences or they may be associated with separate consequences. For example, a general statement of risk of the form:

> there is a lack of coding experience in the development team; code may be inefficient

may be expanded, during analysis and planning activities or as part of continuing risk identification activities, to be

> there is a lack of experienced C programmers; code may be inefficient
>
> and
>
> there is a lack of experienced Ada programmers; code may be late
>
> etc.

The relationships between related risks is shown in Figure 5-3.

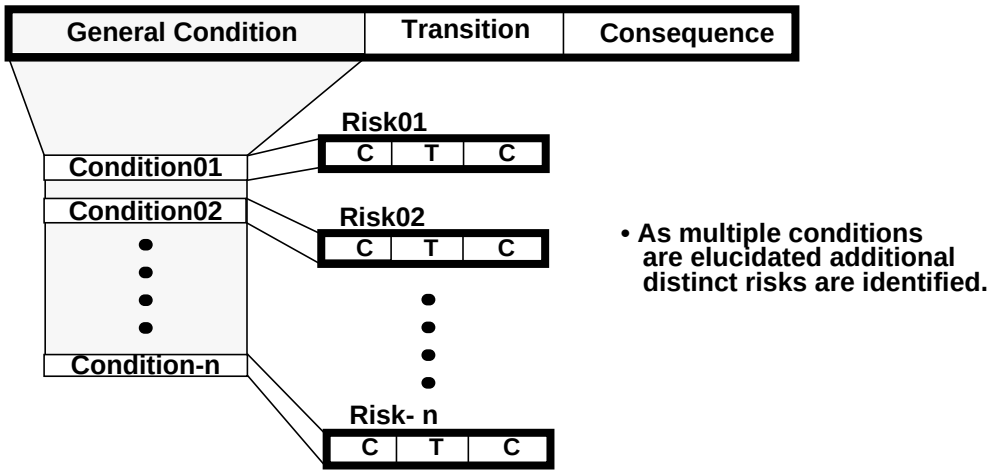


Figure 5-3: Risk Decomposition

It is also possible for distinct, unrelated CTC risks to share a common consequence. For example consider the following risks:

> the development team has never coded an X windows user interface; the code may be late
>
> there are only five workstations for ten development team members; the code may be late
>
> the commercial vendor of the compiler is two weeks late with their delivery of the new version of their Ada compiler; the code may be late

The relationship between multiple risks and a common consequence is shown in Figure 5-4.

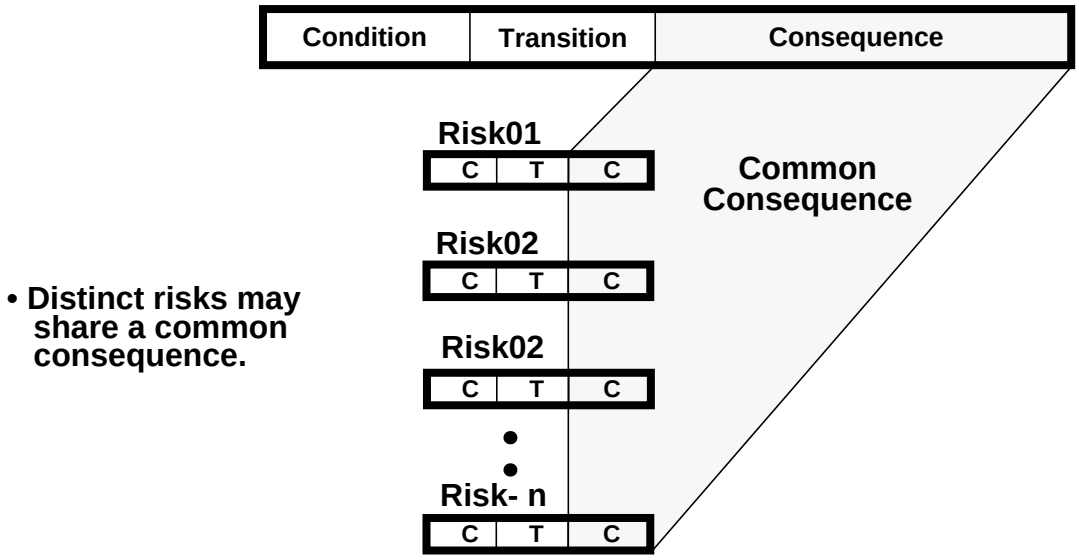


Figure 5-4: Risks Sharing a Common Consequence

6 Risks and Tasks

Often a task within a software-dependent development program is expressed in terms of the activities needed to achieve an outcome. For example, a task in a program plan may be:

> Develop an I/O driver

Lower level tasks may be identified but they can all be expressed in a similar fashion:

> Develop an interrupt handler
> Develop an illegal value handling routine
> etc.

A perspective which can be applied to the description of a task is one of transition (time) and value. The identified result is a desired result; the activities comprising the task accomplish a transition of the system to a new state. As work on the task is completed the system progresses through the program space until all the desired consequences are achieved. Therefore, based on this abstraction, a task can be written as a Condition-Transition-Consequence (CTC) construct and stated, for example, as:

> Given that we do not have an I/O driver that meets our needs [Condition] we must design and code one [Transition], then we will have a system that has one [Consequence].

Thus, tasks involve transition, e.g. code, design, build, and evaluate, and involve action that results in change. Specifically, a task involves taking some action that will result in a desirable consequence. A task then could be generally expressed, albeit awkwardly, as:

> Condition - Transition - Consequence [CTC]
>
> {Given....} then {Do}... then... {consequences}.

The important points here, based upon the abstraction, are that the consequence is significant, the initial condition, although important, is often implicit (if we had an appropriate working I/O driver then no need for the task) and that in all cases a change (transition) must occur. In this case the transition is perceived to be possible, even likely.

Contrast this with the statement of a risk in which the initial conditions are the most significant element and the consequences often are implicit. In either case, though, a transition is required and this transition is perceived to be possible.

In the abstraction three elements are present and important to completely define the task or risk:

1. Condition (initial state)
2. Transition (perceived possible trajectory/path)
3. Consequences (final state)

Both a risk and a task can be characterized by the perception that a transition can occur (time) where the consequence(s) has some associated value. In terms of a state-space model, a task is a potential transition from the current state of the system to some other desirable state.

Figure 6-1 is a representation of the common perspective of the Condition-Transition-Consequence (CTC) construct for both risks and tasks. The contrast in emphasis within the common construct between a risk and a task is shown. In the case of a risk the emphasis (minimal statement) is the condition portion whereas in a task statement the condition is often implicit. The focus for a task is the transition (Do) and the consequence (What will result).

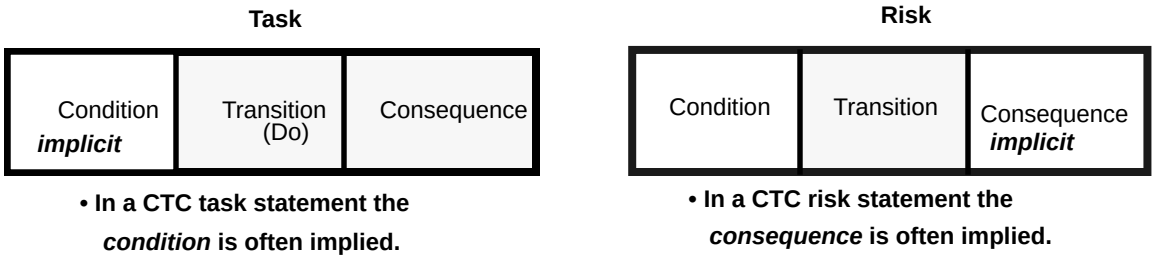


Figure 6-1: The CTC Construct for Risks and Tasks

As the planning effort and the program evolve, existing tasks can be decomposed into other tasks. Similarly risks can be expanded into other risks and additional tasks can be created as a result of the existence of the risk.

Tasks like monitor, observe, and review may have an implied consequence statement. In this case the consequence state may be defined by the criteria establishing when the task is completed. For example, a task such as “monitor the progress of the subcontractor in developing the graphics interface” has an implied completion (consequence) state characterized by the subcontractor completing a project. Often though for these tasks there is an explicit outcome defined, as in the case of a task defined as “monitor the status of the test plan and generate a report.” Thus, while most tasks emphasize the consequence, the transition description can be more important in a task statement than in a risk statement.

7 The CTC Triangle

The common representation of tasks and risks as CTC constructs has implications in the risk identification and task definition (planning) processes. This relationship is shown graphically in Figure 7-1.

The definition of a task, generally part of the planning process, can be viewed somewhat heuristically as first establishing exactly what needs to be developed, i.e. answering the question: *Where do we want to go?* and then making an assessment of what exists now, i.e. answering the question: *Where are we now?* This information forms the basis for defining the method of getting to the final state, that is, answering the question: *What should happen?* (How can this be accomplished?) by relying on the knowledge of both the desired outcome and the current situation.[1] This perspective is one of first establishing the desired (planned) consequences, then identifying the conditions that exist now, and finally defining the transition steps required to achieve the desired consequences.

In contrast, given the CTC-based approach, risk identification initially focuses on the conditions (circumstances) that exist and then involves assessing the potential for adverse consequences based upon a simultaneous consideration of the desired outcome and the current situation. Through the recognition of the conditions the questions: *What may happen? (that may lead to adverse consequences)* and *Where do we not want to go?* can be addressed. This is very much in the sense that a task is defined by simultaneously addressing the desired outcome and the current situation. For example, if one is to address the risks associated with a project intended to develop a high performance real-time graphics software system, the conditions that exist now are particularly important. If the staff assigned the project has never coded a graphics package, one has a different “sense” of the risk compared to a situation where the project has been assigned to a staff of software engineers that has successfully coded three graphics software packages. Thus, in identifying a risk in the CTC representation the starting point is to look at the current (existing) conditions of the project (recognizing the planned outcomes) and to consider the potential transitions of the program to various adverse states (consequences) based upon this information.

1. This perspective involves viewing the planning process for software-dependent development programs as a general problem solving process, one similar to that defined by Simon [Simon 77].

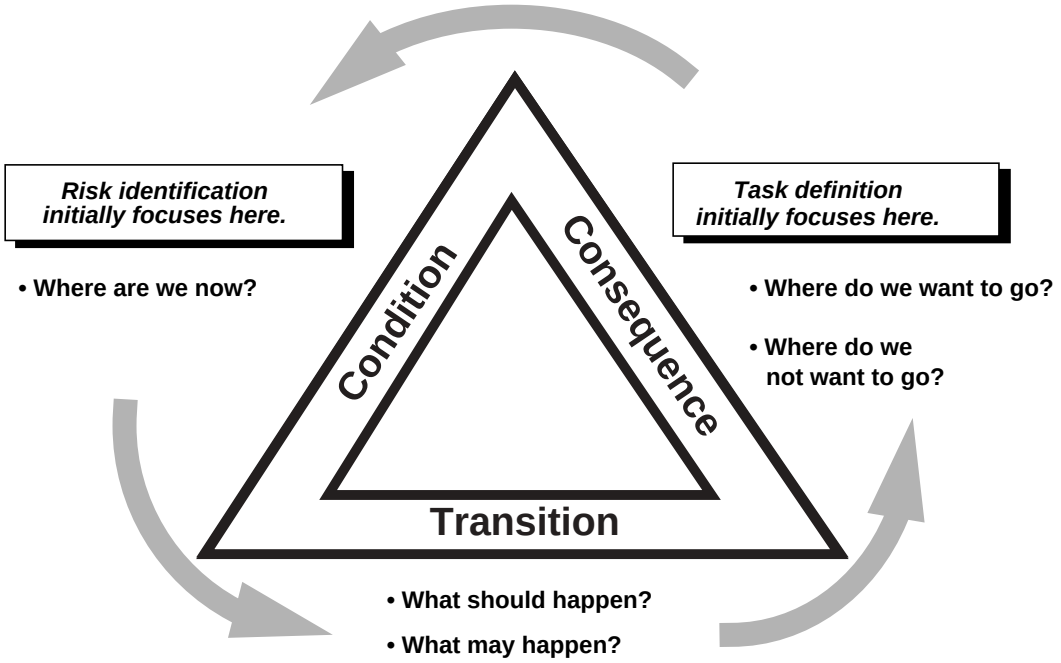


Figure 7-1: The CTC Triangle

The actual thought processes for the identification of risks and the definition of tasks are more complex and more concurrent than suggested by this paradigm. The issue of human ability to accomplish even the most basic tasks involves multiple sources of knowledge, experience, and multiple simultaneous constraints [McClelland 86]. As presented here the risk identification and task definition processes involve significant elements of problem solving (planning) [Simon 83]. The success in addressing these challenges is intimately coupled to the knowledge, training, and experience of the professionals involved in the program. The paradigm represented by the CTC Triangle is intended to be a heuristic, scalable, sequential process structure – a heuristic framework and a practical guide to facilitate the process of identifying program risks.

8 The CTC Approach and Risk Related Concepts

In this section the relationship of the CTC approach to more general risk terminology and risk-related concepts is discussed. The intent is to provide insight into the CTC-based approach by comparing the CTC perspective to other perspectives on risk and risk management.

8.1 CTC Risks and Hazards

The definitions of hazard as a noun are [Houghton 85]:

1. A chance; accident.
2. A chance of being injured or harmed; danger: *Space travel is full of hazards.*
3. A possible source of danger; a fire hazard.
4. A dice game similar to craps.
5. A sandtrap or other obstacle on a golf course.

While the definition for hazard suggests a very close if not overlapping meaning with risk, as noted by Trudy Bell in *IEEE Spectrum* [Bell 89], generally experts distinguish between risk and hazard and employ the following definitions:

> *Hazard*: an intrinsic property or condition that has the potential to cause an accident.
>
> *Risk*: the combination of the probability of an abnormal event or failure and the consequence(s) of that event or failure to a system's operators, users, or its environment.

The CTC definition of a risk includes the probability as an attribute of the CTC risk entity and the consequence(s) as part of the statement. Thus, the definition of risk shown above is completely included in the CTC risk entity.

A hazard in the context of the CTC definition of a risk is a factor within the statement of condition (circumstances) and may constitute the complete statement of the condition. Hazards may be explicitly stated and/or may be included in the *context* attribute of the risk.

As an example consider the definition of risk as a sandtrap. This is one with which many will identify. A sandtrap in the fairway affects the probability of achieving a par score on that hole or, depending upon skill level, on other holes as well. Similarly, a hazard in a software development process may be the limited performance of the development workstations on a project where the risk might be stated as:

> Given the use of very slow development workstations there is concern that (possibly) the software will not be completed on time.

As with the golf example where the trap never affects play, the project may very well get completed successfully but there is the increased possibility of an adverse impact because of the "hazard."

Also consider the example:

> Given that we are using assembly language there is concern that (possibly) the code will not be reliable.

In this case assembly language may or may not be considered by most software experts as a hazard, in and of itself, but it is perceived that its use for the code to be developed entails risk. Thus, the hazardous characteristics associated with assembly language are very subjective and implicit in the sense that more details, generally provided in the context attribute of the risk, are required to fully define the hazardous aspects of the condition statement within the risk. For example the context may be: "You know how hard it is to document and code complex routines in assembly language. The traceability is difficult and no one has coded in that processor's assembly language before."

The CTC construct generally includes hazard(s), as defined above, as part of the statement of a risk.

8.2 CTC Risks and Concerns

In identifying risks, the dimension of value is evidenced by the expression of concern in the CTC statement of risk (*concern* in this context is used in the sense that concern is "anxiety or worry" [Houghton 81]). The expression, "there is concern that" establishes negative associations (loss) with the conditions that characterize a risk in the program. These associations are based upon a judgment (perception), by one or more individuals, on the nature of the possible transition and consequence, the potential outcome(s) of the condition.

In establishing the perception of potential loss, there is a mental extrapolation that involves the possible evolution of the system through time (transition) that culminates in or has as integral points on the transition path, states that involve loss or diminished program success. The details and variation in these perceptions are founded in the knowledge and experience of the individual(s) expressing the concern, those identifying risks. These individual judgments, expressions of concern, regarding transition and consequence establish the negative sense of value (potential loss) for risk in general and for CTC statements of risk specifically.

8.3 A CTC Risk Versus a Problem

There is often an issue relating to whether a situation (condition) is a problem or a risk. Two noteworthy observations are:

- A problem involves a value judgment made upon the merits of the current condition. It is a condition that exists and is undesirable.
- A risk involves a value judgment made upon the potential implications of the current condition. It suggests a possible, future undesirable condition (consequence).

A sense of risk is present as long as there is a perception that the current circumstances may result in loss or harm but for the purposes of identifying and managing software risks, in the CTC representation, a risk is minimally defined as the description of the current condition and a sense of potential loss. This sense of loss may be presented as a notional description of the potential outcome/consequence, i.e., The current circumstances will result in a continuous sequence of states of the system (events) such that at some point in the evolution of the system or as a final consequence, the system will be in an undesirable state.

A risk is a condition that has the potential for undesirable consequences. Simply defining the condition for a risk and expressing the feeling of concern or expressing a notional description of potential adverse consequences (e.g. something bad will happen) makes implicit the exact nature of the loss or harm and constitutes a minimal statement of a CTC risk.

A problem is the existing condition that has undesirable attributes. The loss associated with a problem is generally evident in the description of the condition (the problem). When the condition is described the negative aspects are evident and a sense of undesirable circumstances (loss) exists.

Many problems are risks themselves in that they may lead to more serious symptoms or other problems and diminished success (loss) for the program. This overlap is shown in Figure 8-1.

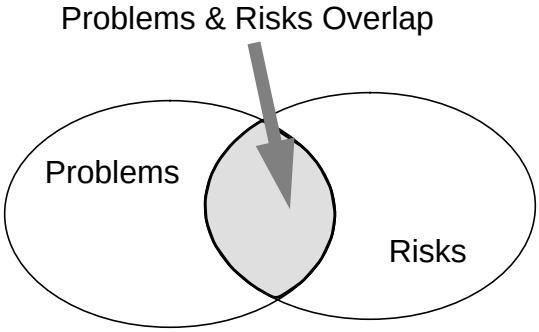


Figure 8-1: Intersection of Set of Problems and Set of Risks

A difference between a problem and a risk is a matter of degree – the extent to which the program is being adversely affected -- and time. The conditions of a problem are more noticeably affecting the program now. A risk, which is also a problem (as represented in the overlap region of Figure 8-1), involves a condition that has a noticeable adverse effect on the program now, but also is perceived to portend additional or more serious problems in the future.

8.4 A CTC Risk and Cause

The issue regarding whether the condition can be viewed as a “source of risk” relates to an assessment of cause. In a deterministic- or even in a probabilistic-based environment the condition can be identified as the source (cause) of the consequence. For example, analyzing the

interaction of a collision between solid objects involves well-defined dynamics that characterize the evolution in time. The correct application of these principles will result in an accurate prediction of the outcome of the collision (the consequence).

In contrast, to say that a condition is a source (cause) of the risk requires not only that the circumstances be described but also the perception that they may lead to negative consequences. Thus, a value judgment must be made by an individual or group of individuals to establish risk and specifically a risk. From this perspective the source as defined above is an integral part of risk and pragmatically is a key defining element of a risk; it is part of the risk. No causal relationships are involved directly with the risk, rather causal issues are related to the temporal aspects of the evolution of the program in time.

One may say that the conditions cause the possibility but this avoids the perception issue relative to value. There is a cause-effect relationship between the time now (condition) and what may occur in the future (consequence). There is also a causal relationship between the condition and the sense of concern, doubt, anxiety, or uncertainty, etc. Collectively these causal relationships are involved with risk and are integral to risk and the definition of a risk in the model.

Thus, within the risk itself there is a sense (concern) relating to the conditions potentially causing a consequence, with the potential outcomes defined by the dynamic characteristics of the environment. These are perceptions of the environment; and risk involves perceptions of individuals who rely on previous experience and their intrinsic characteristics, that are independent of the dynamics of the environment.

It may be asserted that these individuals contribute to the evolution of the program in time, not directly through their perception, but rather through their action or inaction. Again, these may be premised upon perceptions, where these perceptions result in assessing the worth of the results, final or intermediate, that occur and in trying to achieve certain outcomes based upon individual desires, needs, and perceptions. Ultimately, the influence of individual human perceptions is an integral part of cause in risk.

From this perspective the traditional concept “source of risk” as described above, is only a partial statement of the factors that contribute to establishing risk. Other elements of risk relate to the perceptions, values, and characteristics of the individuals involved.

8.5 A CTC Risk Versus Kaplan and Garrick’s Set of Triplets

In the inaugural edition of the journal, *Risk Analysis*, Kaplan and Garrick [Kaplan 81] suggest a quantitative definition of risk that consists of a set of triplets represented as $<s_i, p_i, x_i>$. The triplet is formed by answers to the questions [Kaplan 81]:

1. What can happen? (i.e. What can go wrong?) - The answer is represented as s_i which is a scenario identification or description.
2. How likely is it that that will happen? - The answer is represented as p_i, which is the probability of the scenario.
3. If it does happen, what are the consequences? - The answer is represented as x_i which is the consequence or evaluation of that scenario, i.e., is the measure of damage.

In comparing the $<s_i, p_i, x_i>$ triplet with the CTC construct, CTC provides a structure for expressing the scenario, s_i, where CTC focuses on the initial conditions of the scenario. The triplet scenario addresses the transition-consequence portions of CTC. In general, the CTC approach differentiates risks based upon the characteristics of the current state (condition) and admits the possibility of multiple outcomes (consequences). The sense of the consequence portion of the CTC construct is essentially the outcome in a direct answer to the question: What can happen? In contrast, consequence, as represented by x_i in the triplet, is synonymous with the impact of a CTC risk. In the CTC representation x_i becomes an attribute of the CTC risk entity. Similarly, the probability of a scenario, p_i, is an attribute of the CTC risk entity. The mapping between the triplet and a CTC representation is shown in Figure 8-2.

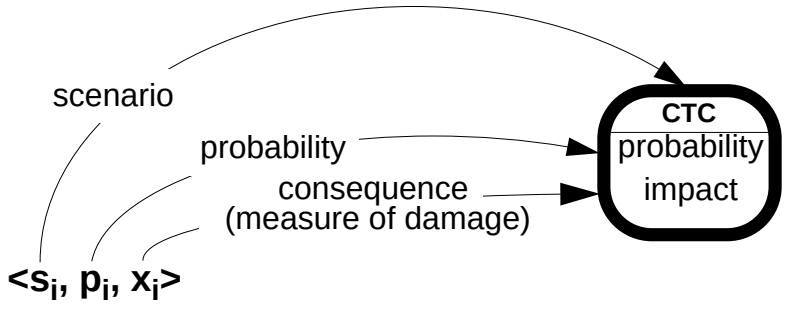


Figure 8-2: Mapping Between a Risk Triplet and a CTC Representation

The contrast between the triplet representation and the CTC approach mirrors the contrast in perspectives evident in the application of risk analysis. In more traditional risk analyses, hardware is the focus with the processes affecting the risks or contributing to increased hazard levels. Risk identification, analysis, and management of software-dependent programs involves the assessment of the potential evolution in time of a complex, dynamic, and abstract system and its associated outcomes. Thus, development program risk analysis investigates risks associated with a possibly controllable evolutionary process, wherein hardware (tools) supports the evolution to a final tangible outcome (product).

9 Summary

As embodied in the Condition-Transition-Consequence (CTC) construct, the CTC approach establishes a distinct, identifiable risk entity, with characteristic risk attributes, that has a life cycle of creation, existence, change, and extinction. In the risk identification process, the CTC format provides structure and guides expression. A CTC risk entity, when embodied with its characteristic attributes, provides sufficient information and structure to enable risks to be effectively managed in subsequent steps of the SEI risk management paradigm [SEI 92].

The CTC concept is a temporal and value-based abstraction that relies upon individual or collective perception. The negative sense of value assigned to a CTC risk differentiates it from a program task, which can also be represented by a CTC construct. The uncertainty associated with risks as well as with tasks is inherent in the kinematics and dynamics of both the program and its environment.

A CTC risk is minimally defined as a description of the current conditions coupled with an expression of concern (potential loss) regarding those conditions. The expression of concern may be embodied in a value judgment within the statement of the condition. While this description of the conditions that have engendered concern has in the past been identified as the source of risk, the focus of risk identification should not be on source or root cause issues but rather on elucidating the significant characteristics of the current circumstances, the conditions. The consequence portion is considered optional during the risk identification process and can be expressed as notional. Thus, once the condition is defined (in the CTC statement as well as in the context) and at least a notional sense of the consequence(s) is established, a risk has been identified.

As applied within a software-dependent development program, the CTC framework would enable program personnel to construct a risk entity and subsequently manage this entity throughout all steps of the SEI risk management paradigm. Specifically, this process would be initiated through the generation of a CTC statement of risk and its context – risk identification. The analysis of root causes, collection of consequences, impacts, probabilities, etc., and the establishment of the set of attribute values and changes in those values would be accomplished as the risk is managed according to the paradigm. This entire process would be an integral part of the overall program management methods employed by the program.

In the CTC approach risks and tasks are viewed as having common characteristics in a common framework: condition-transition-consequence. Through this common perspective, heuristic risk identification and task definition processes, which involve the systematic addressing of questions such as: Where do we want to go?, Where are we now?, What should (may) happen?, and Where do we not want to go?, can be established. In considering these questions, the CTC structure and framework enables the definition of specific risks and specific tasks, and the establishment of the initial values for many of their respective attributes. The distinct entities (tasks and risks) resulting from these initial activities are then managed throughout the life cycle of a program.

Acknowledgments

The author thanks the following members of the team risk management project: Audrey Dorofee, Ron Higuera, Dick Murphy, Julie Walker and Ray Williams for their review of this work; and expresses appreciation to Yacov Haimes, William Harding, Suresh Konda, and Ira Monarch for providing valuable suggestions on this effort. The excellent work of Julia Deems in reviewing this document and the efforts of Sandra Bond, Darlene Bigos, and Jodi Straitiff in its preparation are greatly appreciated.

References

[American 85] *The American Heritage Dictionary: Second College Edition*. Boston: Houghton Mifflin, 1992, 1985.

[Barsalou 92] Barsalou, Lawrence W., "Frames, Concepts, and Conceptual Fields," 21-74. In the book edited by Lehrer, Adrienne; & Kittay, Eva Feder(Eds.). *Frames, Fields, and Contrasts: New Essays in Semantic and Lexical Organization*. Hillsdale, N.J.: L. Erlbaum, 1992.

[Bell 89] Bell, Trudy E. "Managing Murphy's Law: Engineering a Minimum-Risk System." *IEEE Spectrum* 26, 6 (June 1989): 24-27.

[Boehm 91] Boehm, Barry W. "Software Risk Management: Principles and Practices." *IEEE Software* 8, 1 (January 1991): 32-41.

[Carr 93] Carr, Marvin; Konda, Suresh; Monarch, Ira; Ulrich, Carol; & Walker, Clay. *Taxonomy-Based Risk Identification* (CMU/SEI-93-TR-6). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1993.

[Charette 89] Charette, Robert N. *Software Engineering Risk Analysis and Management*. New York: McGraw-Hill, 1989.

[Charette 90] Charette, Robert N. *Application Strategies for Risk Analysis*. New York: Multiscience Press, 1990.

[Chittister 93] Chittister, Clyde; & Haimes, Yacov Y. "Risk Associated with Software Development: A Holistic Framework for Assessment and Management." *IEEE Transactions on Systems, Man, and Cybernetics* 23, 4 (1993).

[Gardner 85] Gardner, Howard. *The Mind's New Science*. New York: Basic Books, 1985.

[Gove 81] Gove, Phillip Babcock, Editor. *Webster's Third New International Dictionary: Unabridged*. Springfield, MA: Merrian-Webster, 1981.

[Haimes 89] Haimes, Yacov Y. "Total Risk Management." *Risk Analysis* 11, 2 (1991): 147-149.

Haimes 91] Haimes, Yacov Y. "Toward a Holistic Approach to Risk Assessment and Management." *Risk Analysis* 9, 2 (1989): 147-149.

[Henley 92] Henley, Ernest J.; & Kumamoto, Hiromitsu. *Probabilistic Risk Assessment*. New York: IEEE Press, 1992.

[Higuera 93] Higuera, Ronald, P.; & Gluch, David P. "Risk Management and Quality in Software Development." Paper presented at the Eleventh Annual Pacific Northwest Software Quality Conference, October 18-20, 1993.

[Kaplan 81] Kaplan, Stanley; & Garrick, John B. "On The Quantitative Definition of Risk." *Risk Analysis* 1,1 (1981): 11-27.

[Kirkpatrick 92] Kirkpatrick, Robert J.; Walker, Julie A.; & Firth, Robert. *Software Development Risk Management: An SEI Appraisal* (SEI Technical Review 92). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1992.

[Levy 93] Levy, S.; Subrahmanian, E.; Konda, S.; Coyne, R.; Westerberg, A.; & Reich, Y. *An Overview of the n-dim Environment*. Technical Report EDRC-05-65-93, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA: 1993

[Lowrance 76] Lowrance, W. W. *Of Acceptable Risk*. Los Altos, CA: William Kufamann, 1976.

[McClelland 86] McClelland, J. L.; Rumelhart, D. E.; & Hinton, G.E. "The Appeal of Parallel Distributed Processing." *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: The MIT Press, 1986.

[Rowe 88] Rowe, William, D. *An Anatomy of Risk*. Malabar, FL: Robert E. Krieger, 1988.

[SEI 92] Software Engineering Institute. "The SEI Approach to Managing Software Technical Risks."*Bridge* (October 1992):19-21.

[Simon 77] Simon, Herbert A. *The New Science of Management decision.* Englewood, NJ: Prentice-Hall, 1977.

[Simon 83] Simon, Herbert A. "Search and Reasoning in Problem Solving." *Artificial Intelligence*, 21 (1983): 7-29.

[Spectrum 89] "Managing Risk in Large Complex Systems." *IEEE Spectrum* 26, 6 (June 1989): 22-23.

[Taha 87] Taha, Hamdy A. *Operations Research: An Introduction*. New York: Macmillan, 1987.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for Public Release
Distribution Unlimited

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
CMU/SEI-94-TR-14

5. MONITORING ORGANIZATION REPORT NUMBER(S)
ESD-TR-94-014

6a. NAME OF PERFORMING ORGANIZATION
Software Engineering Institute

6b. OFFICE SYMBOL (if applicable)
SEI

7a. NAME OF MONITORING ORGANIZATION
SEI Joint Program Office

6c. ADDRESS (city, state, and zip code)
Carnegie Mellon University
Pittsburgh PA 15213

7b. ADDRESS (city, state, and zip code)
HQ ESC/ENS
5 Eglin Street
Hanscom AFB, MA 01731-2116

8a. NAME OFFUNDING/SPONSORING ORGANIZATION
SEI Joint Program Office

8b. OFFICE SYMBOL (if applicable)
ESC/ENS

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F1962890C0003

8c. ADDRESS (city, state, and zip code))
Carnegie Mellon University
Pittsburgh PA 15213

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO	PROJECT NO.	TASK NO	WORK UNIT NO.
63756E	N/A	N/A	N/A

11. TITLE (Include Security Classification)
A Construct for Describing Software Development Risks

12. PERSONAL AUTHOR(S)
David P. Gluch

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (year, month, day)	15. PAGE COUNT
Final	FROM TO	July	47

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (continue on reverse of necessary and identify by block number)

19. ABSTRACT (continue on reverse if necessary and identify by block number)

This report establishes a representation of software risk wherein the risks associated with software-dependent development programs are defined as distinct, manageable risk entities. The risk entities and their descriptive statements of risk are based upon a Condition-Transition-Consequence (CTC) construct. The CTC construct arises out of a systems representation, where time and value are identified as fundamental to the concept of risk. The CTC construct is also shown to provide a common representation for both program risks and program tasks and to fit into a heuristic framework for identifying risks within software-dependent development programs. Examples of risks are used to dem-

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ■ SAME AS RPT ☐ DTIC USERS ■

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified, Unlimited Distribution

22a. NAME OF RESPONSIBLE INDIVIDUAL
Thomas R. Miller, Lt Col, USAF

22b. TELEPHONE NUMBER (include area code)
(412) 268-7631

22c. OFFICE SYMBOL
ESC/ENS (SEI)

ABSTRACT — continued from page one, block 19

onstrate that the approach facilitates the management of risk as an integral part of routine program management.