"The Role of Metrics in Risk Management Across the Software Development Lifecycle"

Dr. Linda H. Rosenberg	Frank Parolek	Steve Botzum
NASA GSFC	Unisys @ NASA GSFC	Unisys @ NASA GSFC
Bld 6 Code 302	Bld 6 Code 304	Bld 6 Code 304
Greenbelt, MD 20771	Greenbelt, MD 20771	Greenbelt, MD 20771
301-286-0087	301-286-0103	301-286-0103
Linda.Rosenberg@gsfc.nasa.gov	Frank.Parolek.1@gsfc.nasa.gov	sbotzum@pop300.gsfc.nasa.gov

Introduction

A vigorous software risk management program is essential in virtually any project undertaken today. The many benefits that risk management brings to a project become doubly important in today's competitive environment. Risk management allows software projects to identify, address and mitigate potential problems early in the development lifecycle, thus allowing time to act and potentially reduce costs and schedule slips. By incorporation this process, a project that embraces risk management concepts and integrates a formalized risk management process into its daily business practices is far more likely to succeed than a project that includes little or no risk management. Add a solid metrics program to that process and you are undoubtedly setting your project up for success.

NASA understands the need for implementing a formal, standardized risk management process in order to meet its Faster, Better, Cheaper (FBC) goal and the Software Assurance Technology Center (SATC) at the NASA Goddard Space Flight Center has been designated as the party responsible for NASA's risk management training. Risk Management is something that, if done properly, occurs on a continual basis; hence, the title of the course: *Continuous Risk Management*. The course was originally taught only at NASA centers, but has now become available to other government agencies and contractors supporting the government. During December 1999, SATC surpassed the 1000-student mark. Project types have included space, ground, hardware, software, shipbuilding, aviation safety, and even business systems.

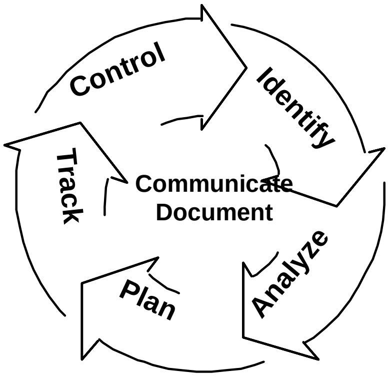


Figure 1: Continuous Risk Management Paradigm

As shown in Figure 1, the course emphasizes the six critical functions of the continuous risk management paradigm: (1) *Identify* the risks in a specific format; (2) *Analyze* the risk probability, impact/severity, and timeframe; (3) *Plan* the approach; (4) *Track* the risk through data compilation and analysis; (5) *Control* and monitor the risk; (6) *Communicate and Document* the process and decisions.

Continuous Risk Management Principle Functions

Identify

The purpose of identification is to consider risks before they become problems and to incorporate this information into the project management process. Anyone in a project can identify risks to the project. Each individual has particular knowledge about various parts of a project. During Identify, uncertainties and issues about the project are transformed into distinct (tangible) risks that can be described and measured.

During this function, all risks are written with the same, two-part format. The first part is the risk statement, written as a single statement concisely specifying the cause of the concern as well as its impact. The second part may contain additional supporting details in the form of a context.

The aim for a risk statement is that it be clear, concise, and sufficiently informative that the risk is easily understood. Risk statements in standard format must contain two parts: the condition and the consequence. The condition-consequence format provides a complete picture of the risk, which is critical during mitigation planning. It is read as follows:

> *given the <**condition**> there is a possibility that <**consequence**> will occur*

The *condition* component focuses on what is currently causing concern; it must be something that is true or widely perceived to be true. This component provides information that is useful when determining how to mitigate a risk. The *consequence* component focuses on the intermediate and long-term impact of the risk. Understanding the depth and breadth of the impact is useful in determining how much time, resources, and effort should be allocated to the mitigation effort. A well-formed risk statement usually has only one condition, but may have more than one consequence. Risk statements should avoid:

- abbreviations/acronyms that are not readily understood
- sweeping generalizations
- massive, irrelevant detail

Since the risk statement is to be concise, a context is added to provide enough additional information about the risk to ensure that the original intent of the risk can be understood by other personnel, particularly after time has passed. An effective context captures the what, when, where, how, and why of the risk by describing the circumstances, contributing factors, and related issues (background and additional information that are NOT in the risk statement).

A diagram of the complete risk statement and context are shown in Figure 2.

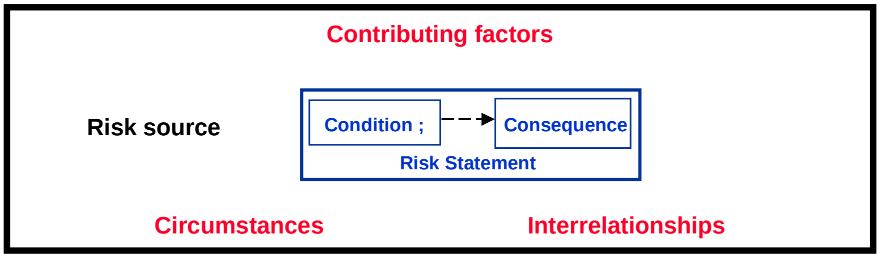


Figure 2: Risk Statement and Context

An example is shown in Figure 3. Note there is one condition and two consequences in the risk statement. The context explains why this is a risk, and supplies additional information for someone unfamiliar with this risk.

<u>Risk statement:</u>

This is the first time that the software staff will use OOD; the staff may have a lower-than-expected productivity rate and schedules may slip because of the associated learning curve.

<u>Context:</u> Object oriented development is a very different approach that requires special training. There will be a learning curve until the staff is up to speed. The time and resources must be built in for this or the schedule and budget will overrun.

Figure 3: Example Risk Statement and Context

Risk identification depends heavily on both open communication and a forward-looking view to encourage all personnel to bring forward new risks and to plan beyond their immediate problems. Although individual contributions play a role in risk management, teamwork improves the chances of identifying new risks by allowing personnel to combine their knowledge and understanding of the project.

Analyze

The purpose of Analyze is to convert the data into decision-making information. Analysis is a process of examining the risks in detail to determine the extent of the risks, how they relate to each other, and which ones are the most important. Analyzing risks has three basic activities: evaluating the attributes of the risks (impact, probability, and timeframe), classifying the risks, and prioritizing or ranking the risks.

Evaluating - The first step provides better understanding of the risk by qualifying the expected impact, probability, and timeframe of a risk. This involves establishing values for:

Impact: the loss or negative affect on the project should the risk occur
Probability: the likelihood the risk will occur
Timeframe: the period when you must take action in order to mitigate the risk

Figure 4 on the next page demonstrates sample values that might be used to evaluate a risk's attributes

Attribute	Value	Description
Probability	Very Likely (H)	High chance of this risk occurring, thus becoming a problem > 70%
	Probable (M)	Risk like this may turn into a problem once in a while {30% < x < 70%}
	Improbable (L)	Not much chance this will become a problem {0% < x < 30%}
Impact	Catastrophic (H)	Loss of system; unrecoverable failure of system operations; major damage to system; schedule slip causing launch date to be missed; cost overrun greater than 50% of budget
	Critical (M)	Minor system damage to system with recoverable operational capacity; cost overrun exceeding 10% (but less than 50% of planned cost
	Marginal (L)	Minor system damage to project; recoverable loss of operational capacity; internal schedule slip that does not impact launch date cost overrun less than 10% of planned cost
Timeframe	Near-term (N)	Within 30 days
	Mid-term (M)	1 to 4 months from now
	Far-term (F)	more than 4 months from now
		NOTE: refers to when action must be taken

Figure 4: Sample Attribute Values

Classifying - The next step is to classify risks. There are several ways to classify or group risks. The ultimate purpose of classification is to understand the nature of the risks facing the project and to group any related risks so as to build more cost-effective mitigation plans. The process of classifying risks may reveal that two or more risks are equivalent—the statements of risk and context indicate that the subject of these risks is the same. Equivalent risks are therefore duplicate statements of the same risk and should be combined into one risk.

Prioritize - The final step in the Analysis function is to prioritize the risks. The purpose is to sort through a large number of risks and determine which are most important and to separate out which risks should be dealt with first (the vital few risks) when allocating resources. This involves partitioning risks or groups of risks based on the “vital few” sense and ranking risks or sets of risks based on consistently applying an established set of criteria. No project has unlimited resources with which to mitigate risks. Thus, it is essential to determine consistently and efficiently which risks are most important and then to focus those limited resources on mitigating risks.

Conditions and priorities will change during a project, and this natural evolution can affect the important risks to a project–. *Risk analysis must be a continual process.* Analysis requires open communication so that prioritization and evaluation is accomplished using all known information. A forward-looking view enables personnel to consider long-range impacts of risks.

Plan

Planning is the function of deciding what, if anything, should be done about a risk or set of related risks. In this function decisions and mitigation strategies are developed based on current knowledge of project risks.

The purpose of plan is to:

- make sure the consequences and the sources of the risk are known
- develop effective plans
- plan efficiently (only as much as needed or will be of benefit)

- produce, over time, the correct set of actions that minimize the impacts of risks (cost and schedule) while maximizing opportunity and value
- plan important risks first

Figure 5 indicates the potential approaches to Risk Planning.

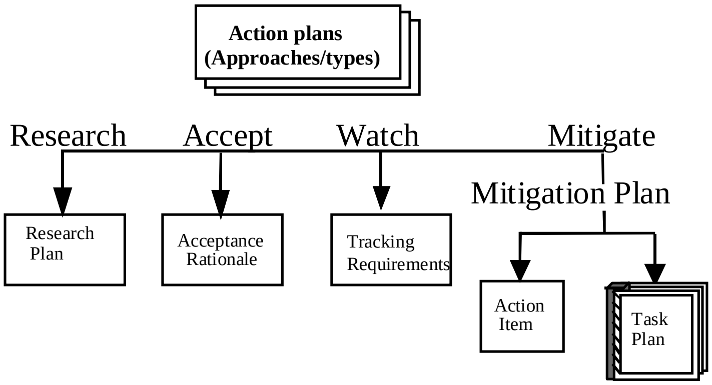


Figure 5: Planning approaches

There are four options to consider when planning for risks:

1. *Research:* establish a plan to research the risk(s)
2. *Accept:* decide to "accept" the risk(s) and document the rationale behind the decision
3. *Watch:* monitor risk conditions for any indications of change in probability or impact (tracking metrics must be established and documented)
4. *Mitigate:* allocate resources and assign actions in order to reduce the probability or potential impact of risks. This can range from simple tasking to sweeping activities:

   Action Items: a series of discrete tasks to mitigate risk
   Task Plan: formal, well documented and larger in scope

Dealing with risk is a continuous process of determining what to do with new concerns as they are identified and efficiently utilizing project resources. An integrated approach to management is needed to ensure mitigation actions do not conflict with project or team plans and goals. A shared product vision and global perspective are needed to create mitigation actions on the macro level to the benefit the project, customer and organization. The focus of risk planning is to be forward looking, to prevent risks from becoming problems. Teamwork and open communication enhance the planning process by increasing the amount of knowledge and expertise that can be applied to the development of mitigating actions.

Track

Tracking is the process by which risk status data are acquired, compiled, and reported
The purpose of Track is to collect accurate, timely, and relevant risk information and to present it in a clear and easily understood manner to the appropriate people/group. Tracking is done by those responsible for monitoring "watched" or "mitigated" risks. Tracking status information become critical to performing the next function in the Continuous Risk Management paradigm, i.e. Control. Supporting information, such as schedule and budget variances, critical path changes, and project/performance indicators can be used as triggers, thresholds, and risk- or plan-specific measures where appropriate.

When a mitigation plan has been developed for a risk or risk set, both the mitigation plan and the risk attributes are tracked. Tracking the mitigation plan, or even a list of action items, will indicate whether the plan is being executed correctly and/or on schedule. Tracking any changes in the risk attributes will indicate whether the mitigation plan is reducing the impact or probability of the risk. In other words, tracking risk attributes gives an indication of how effective the mitigation plan is.

Program and risk metrics provide decision makers with the information needed for making effective decisions. Normally program metrics are used to assess the cost and schedule of a program as well as the performance and quality of a product. Risk metrics are used to measure a risk's attributes and assess the progress of a mitigation plan. They can also be used to help identify new risks.

> *Example*: A program metric might look at the rate of module completion. If this metric indicates that the rate of completion is lower than expected, then a schedule risk should be identified.

Open communication regarding risk and mitigation status stimulates the project and risk management process. Tracking is a continuous process - current information about a risk status should be conveyed regularly to the rest of the project. Risk metrics provide decision makers with the information needed for making effective decisions.

Control

The purpose of the Control function is to make informed, timely, and effective decisions regarding risks and their mitigation plans. It is the process that takes in tracking status information and decides exactly what to do based on the reported data. Controlling risks involves analyzing the status reports, deciding how to proceed, and then implementing those decisions.

Decision makers need to know 1) when or whether there is a significant change in risk attributes and 2) the effectiveness of mitigation plans within the context of project needs and constraints. The goal is to obtain a clear understanding of the current status of each risk and mitigation plan relative to the project and then to make decisions based on that understanding. Tracking data is used to ensure that project risks continue to be managed effectively and to determine how to proceed with project risks. Options include:

– *Replan* - A new or modified plan is required when the threshold value has been exceeded, analysis of the indicators shows that the action plan is not working, or an unexpected adverse trend is discovered.
– *Close the risk* - A closed risk is one that no longer exists or is no longer cost effective to track as a risk. This occurs when: the probability falls below a defined threshold, impact lies below a defined threshold, or the risk has become a problem and is tracked.
– *Invoke a contingency plan* - A contingency plan is invoked when a trigger has been exceeded or some other related action needs to be taken.
– *Continue tracking and executing the current plan* - No additional action is taken when analysis of the tracking data indicates that all is going as expected or project personnel decide to continue tracking the risk or mitigation plan as before.

Open communication is important for effective feedback and decision making - a critical aspect of Control. Risk control is also enhanced through integrated management - combining it with routine project management activities enables comprehensive project decision making.

Communication & Documentation

The purpose of Communicate and Document is for *all* personnel to understand the project's risks and mitigation alternatives as well as risk data and to make effective choices within the constraints of the project. Communication and Documentation are essential to the success of all other functions within the paradigm and is critical for managing risks.

<u>Identify</u>: In risk identification, risk statements are communicated.

<u>Analyze</u>: In analysis, project personnel communicate information about impact, probability, and timeframe attributes. Risk classification involves grouping risk information communicated by individuals.

<u>Plan</u>: During planning, action plans are developed and communicated to project personnel.

<u>Track</u>: Reports designed to communicate data to decision-makers are compiled during tracking.

<u>Control</u>: The decisions made during control must be communicated and recorded to project personnel.

For effective risk management, an organization must have open communication and formal documentation. Communication of risk information is often difficult because the concept of risk comprises two subjects that people don't normally deal well with: probability and negative consequences.

Not only Continuous Risk Management, but the project as a whole, is in jeopardy when the environment is not based on open communication. No one has better insight into risks than project personnel, and *management needs that input*. Experienced managers know that the free flow of information can make or break any project. Open communication requires:

- Encouraging free-flowing information at and between all project levels
- Enabling formal, informal and impromptu communication
- Using consensus-based processes that value the individual voice, bringing unique knowledge and insight to identifying and managing risks.

Introducing Metrics into Continuous Risk Management

A critical aspect of the successful implementation of risk management is introduced in Module 6 of the course – "Track", and is used in subsequent risk management activities. Software metrics are introduced as a technique for monitoring or tracking a risk. Before the importance of metrics in the software development cycle and risk management process can be fully understood, it becomes necessary to answer the question "Why are metrics really necessary?"

- To gauge the success of the development process
- To identify potential problems
- To better evaluate the quality of the product
- To provide meaningful information to decision-makers
- To allow decision-makers to lower cost overruns and make better use of existing resources

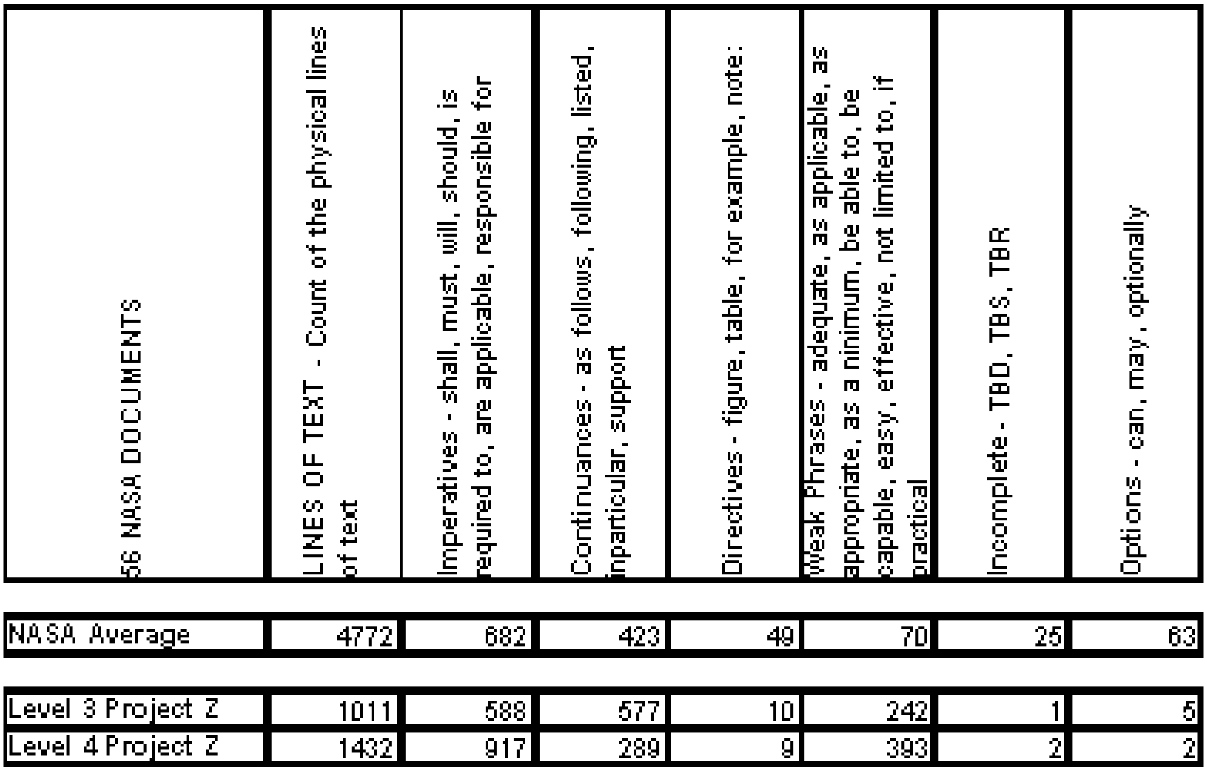
Risk management and metrics are mutually supportive and interdependent and both are very much needed in today's "faster, better, cheaper" environment. These metrics can identify new risks as they provide information pertaining to known risks. In general, metrics are used to assess the attributes of a risk (impact, probability, and timeframe); provide meaningful information in support of informed control decisions; and assess the success of mitigation planning. The remainder of this paper will demonstrate selected metrics for the various phases of the software development lifecycle from requirements through testing.

Metrics Program Goals

A solid metrics program is critical for tracking the status of a risk mitigation or monitoring effort. To that end, the SATC has developed four primary software metrics program goals: 1) Requirements Quality; 2) Product Quality; 3) Test Effectivity; 4) Process Effectivity. Let us look at each of these goals in-depth.

Goal 1: Requirements Quality

The first software metrics program goal is Requirements Quality. Risks that originate in the requirements phase that later become problems and are identified in the testing phase can cost 100 or more times to fix than if they had been identified and fixed in the requirements phase. Therefore, it is imperative to analyze requirements to a point where a reasonable confidence level in those requirements can be attained. Requirements that have multiple interpretations or are unclear, and changes to requirements all represent risk. In an effort to identify and understand these and other risks, the attributes of requirements quality have been defined as Ambiguity, Completeness, Volatility, Traceability, and Understandability.



56 NASA DOCUMENTS	LINES OF TEXT - Count of the physical lines of text	Imperatives - shall, must, will, should, is required to, are applicable, responsible for	Continuances - as follows, following, listed, inparticular, support	Directives - figure, table, for example, note:	Weak Phrases - adequate, as applicable, as appropriate, as a ninimum, be able to, be capable, easy, effective, not limited to, if practical	Incomplete - TBD, TBS, TBR	Options - can, may, optionally
NASA Average	4772	682	423	49	70	25	63
Level 3 Project Z	1011	588	577	10	242	1	5
Level 4 Project Z	1432	917	289	9	393	2	2

Figure 6: Automated Requirements Measurement tool results

Metrics can be used to determine the degree of requirements ambiguity through the measure of weak phrases that convey multiple meanings. Requirements that contain terms such as adequate, as appropriate, easy, etc. do not truly require anything. The ambiguity of the terms leaves room for various interpretations and the possibility that the intended requirements will not be fulfilled.

Completeness of requirements is characterized by the presence of TBDs (to be determined), TBSs (to be specified), etc. Incomplete requirements can have a cumulative effect on subsequent phases of the development cycle, resulting in an exponential growth of unknowns.
The ARM (Automated Requirements Measurement[1]) tool, developed by the SATC, is used to measure both requirements ambiguity and completeness. (See Figure 6.)

Requirements Volatility can be ascertained by examining the number of changes or modifications to requirements. Add to this information the number of new requirements being added and the number of requirements being deleted and you are left with a good picture of the level of volatility of your requirements.

Requirements should be traceable, that is one should be able to trace a requirement both upward to higher level documents and downward to code and tests. By identifying the number of items that cannot be traced either up or down, one gains a better picture of the traceability of requirements.

Requirements should be relatively easily understandable. The smoother a document reads, the easier it will be to understand. This can be measured by examining the depth of the numbering scheme and the readability index of the requirement.

Goal 2: Product Quality

The second software metrics program goal is Product Quality. There are various risks associated with the product quality goal, which include but are not limited to: large complex modules with poor branching, inadequate or out of date documentation, a system that is costly to maintain, and a system that is difficult to reuse. To understand and organize risks associated with product quality, the attributes have been defined as Structure/Architecture, Reuse, Maintainability, and Documentation.

In order to determine the integrity of the structure and architecture of a software module, an evaluation of the constructs within that module should be performed to identify possible error-prone modules and to indicate potential problems in usability and maintainability.

It is sometimes beneficial to reuse software modules rather than write a module from scratch, so it is helpful to examine the suitability of software for reuse in a different context or application. We can assess the potential for reuse of a given module by examining its complexity. The more complex a module, the less likely it is to be a good candidate for reuse. At the same time, complexity can be a good indicator of maintainability. Software that is difficult to maintain and correct can be cumbersome and waste of valuable resources. The suitability of the software for ease of locating and fixing a fault in the program should be determined. For example, during the coding phase, metrics reflective of cyclomatic complexity can be collected on the files and

[1] ARM is available free from the SATC web site at "http://satc.gsfc.nasa.gov"

subsystems. Using a size indicator, such as lines of code or executable statements, the modules can be plotted on a scatter plot to determine the maintainability of the modules. An example of this is shown in Figure 7. Using this data, we can easily identify the modules of code that are outside the expected ranges, hence, identifying potential risks to maintainability and reuse.

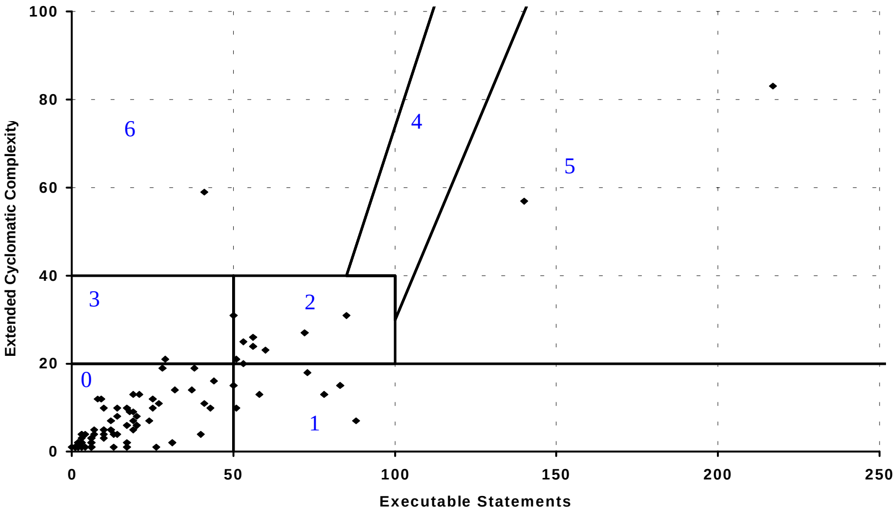


Figure 7: Example of Code Metrics Application

The modules of code in the areas labeled 0, 1, 2, 3 are low risk. The code in the other sections is at higher risk. These areas were identified by the SATC based on several NASA projects.

The adequacy of internal code documentation and external documentation should also be determined to gain a better understanding of the supportability of that software.

Goal 3: Test Effectivity

The third software metrics program goal is Test Effectivity. There are various risks associated with the test effectivity goal, which include but are not limited to the inability to locate and repair expected number of faults, tests not completed on schedule, error concentrations in specific software segments, and insufficient or excessive test coverage to verify functionality. To understand and organize risks associated with test effectivity, the quality attributes have been defined as Correctness, Error Concentrations, Comprehensiveness, and Completion Rates.

To evaluate the correctness of software we need to look at how free that software is of faults. Determination of the localization of errors and high criticality errors in specific software segments helps us to understand our second attribute of Error Concentrations. Comprehensiveness can be better understood by determining the amount of coverage and a measure of completion rates will give a good indicator of how many tests are being completed on schedule. The combination of these four quality attributes allows us to better understand overall test effectivity.

Most projects are familiar with testing metrics, tracking the cumulative number of problem reports (defects) that are open versus the number closed. This data is very helpful in monitoring the schedule for the testing phase, for determining if testing will be completed within the estimated time, or more specifically, if all problem reports will be closed within the remaining time frame.

Metrics can also be used to identify possible risks inherent in the software development process itself. A common metric used during all phases is the rate of component completion. In the requirements phase, it is common to refer to the number of requirements completed; in the coding phase, attention shifts to the number of modules completed; and in the testing phase, the focus is on the number of tests completed. The same type of graph can be used for all phases as shown in Figure 8.

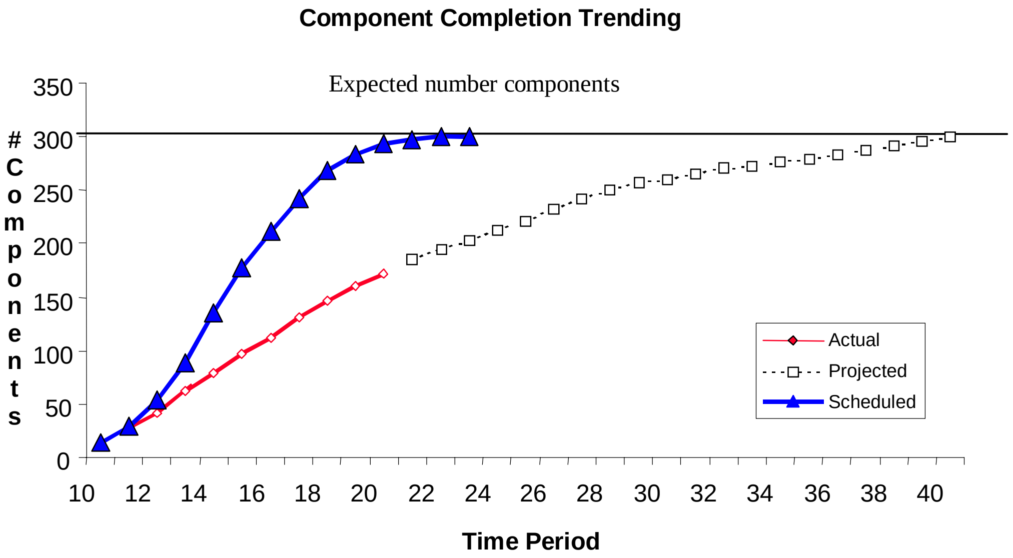


Figure 8: Example of Component Tracking Data

In Figure 8, the number of components projected versus the number actually completed is tracked. The deviation noted as a risk at week 14 has become a significant problem by week 21, when it becomes obvious that this project's schedule is in trouble. Were these metrics not tracked and monitored from the beginning, the risk at week 14 would have gone unnoticed. At week 21, the situation has grown to a serious problem from which recovery may not be possible. These data demonstrate the importance and value of risk management as well as the critical role that metrics can play in project management. A variation may be fixable when detected early. Careful tracking and monitoring is the means to that end.

Goal 4: Process Effectivity

The fourth software metrics program goal is Process Effectivity. Appropriateness of activities and slow completion rates are just two risks that are faced when attempting to determine process effectivity. To understand and organize risks associated with process effectivity, the quality attributes have been defined as Resource Use and Completion Rates.

When a project is in the requirements phase, it would be common sense to assume that all activities in that phase are related to requirements. The same is applicable to later phases of the lifecycle. All too often, the lines between the various development phases are blurred to the extent that it is difficult to determine where one phase ends and the next begins (see figure 9 next page). A determination of the extent that resource usage correlates to the appropriate phase of the project will assist in the understanding of risk associated with that blurring. Another aspect that needs to be examined is the progress in completing items such as peer reviews and the turnover of completed modules to Configuration Management (CM).

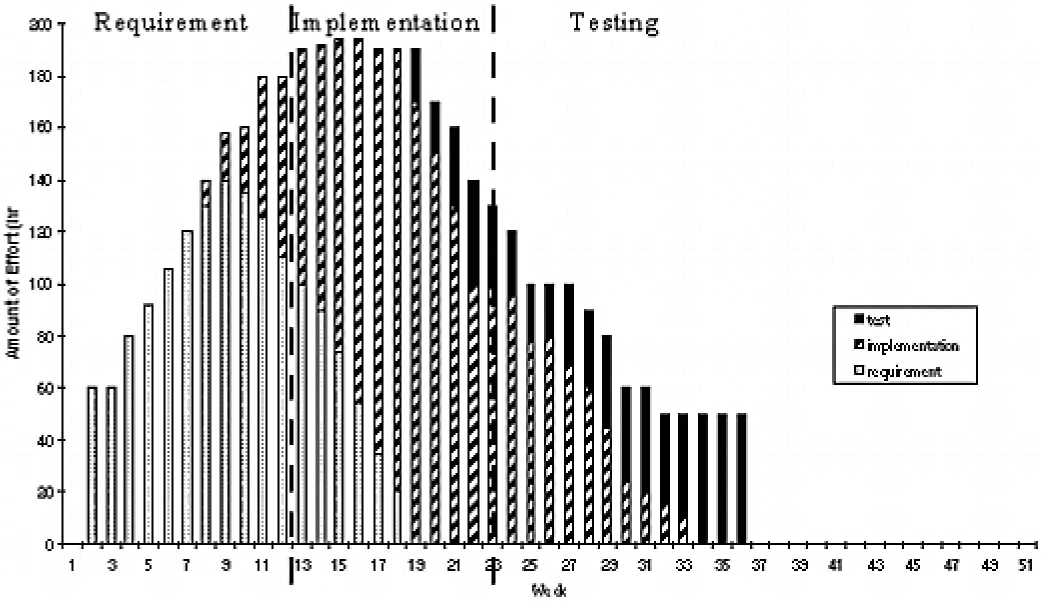


Figure 9 - Effort Per Activity

To assist us in better understanding our resource use and completion rates there are various metrics we can utilize. We can look at task completions and scheduled task completions, staff hours spent on life cycle task activities, the given lifecycle phase as compared to the scheduled lifecycle phase, etc.

Bringing It All Together

One should always keep in mind the purpose of gathering these metrics in a Continuous Risk Management context. Metrics are used to identify risks, to monitor the implementation of a mitigation strategy, and to determine the effectiveness of a given mitigation strategy by measuring the effect of that strategy on the probability and impact of a risk. The choice of which metrics to gather depends on what is an appropriate indicator of change in the status of a risk or for information gathering which could also be used to identify risks.

When determining which metrics to use, a few simple guidelines are sufficient:

1. The metrics should be simple to understand. If they are difficult to explain how to compute, interpret the results or use, they are too complex.
2. There should be clear, precise definitions and equations. There should only be one definition that everyone uses and understands.
3. They should be as objective as possible; there should not be multiple interpretations.
4. The metrics should be cost effective. A metric that is expensive to collect will probably result in fewer metrics being collected. The metrics should not cost more than the savings that may result.
5. The purpose of the metrics program is to help improve the software, therefore, the metrics should help the project manager or developers improve the quality of the final product.

The preceding are just a few examples of how a well-crafted metrics program, molded together with a well-balanced risk management program, can improve your overall development process. A well-planned metrics collection scheme can yield the information necessary to ensure that you are informed of risk status, to support effective mitigation planning, and to provide pertinent data needed by decision-makers. Metrics need not be expensive to collect; in fact, the most effective metrics are often based on existing activities. Even with some up front investment costs, proper metrics collection and risk management have the potential of realizing sizeable dividends throughout the life of a project. The existence of a dynamic risk management program within the software development lifecycle is a proactive approach to identify and manage risks, but it cannot be effective without supporting metrics to track risk status and the effectiveness of mitigation strategies.

References

Rosenberg, Linda; Gallo, Albert; and Parolek, Frank; "Continuous Risk Management Structure of Functions at NASA"; American Institute of Aeronautics and Astronautics; Albuquerque, NM; 1999.

Audrey J. Dorofee, Julie A. Walker, and others; "Continuous Risk Management Guidebook"; Carnegie Mellon University, Software Engineering Institute; 1996.

Rosenberg, Linda; and Gallo, Albert; "Metrics for Quality Assurance"; 9th International Conference on Software Quality; 2000.

Rosenberg, Linda; and Gallo, Albert; "Software Metrics"; Intelligent Synthesis Environment - New Millennium; 2000.